Journal of Information Security and Applications 73 (2023) 103444

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

A novel 5-bit S-box design for lightweight cryptography algorithms

Vishal A. Thakor [a], Mohammad A. Razzaque [a,*], Anand D. Darji [b], Aksh R. Patel [b]

[a] School of Computing, Engineering and Digital Technologies, Teesside University, UK
[b] Department of Electronics Engineering, Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India

ARTICLE INFO

Keywords:
Internet of Things (ioT)
Radio Frequency Identification (RFID)
Lightweight cryptography (LWC)
Substitution-box (S-box)
Chaotic mapping
Cryptanalysis
ASIC platform

ABSTRACT

Cryptography is one of the techniques to secure communication and data transfer over the network. It performs well on resource-rich devices (PC, servers, smartphones, etc.). However, it may not fit or, if forcefully fitted, perform poorly on the resource-constrained Internet of Things (IoT) devices (e.g., Radio Frequency Identification (RFID) tags, sensors). For these reasons, there is a need for a lightweight version of cryptography, called lightweight cryptography (LWC). While designing any cryptography algorithm, a substitution box (S-box) is a core and the only component that offers a nonlinear functionality between inputs and outputs. Various researchers propose various S-box designs for different applications. Still, very few of them maintain the trade-offs among cost, performance and security, especially when considered resource-constrained IoT devices. First, the article discusses various S-boxes used in the popular LWC algorithms by their input–output bit-size (3/4/5/6/8 bit) and highlights their strengths and limitations. Then, it focuses on the proposed 5-bit S-box design. The novel design uses a chaotic mapping theory to offer a random behaviour of the element in the proposed S-box. The experimental results from ASIC implementation reveal two essential characteristics of the proposed S-box, cost and performance, and further, compare it with 4/5-bit S-box competitors. Finally, the article demonstrates the security strength of the proposed 5-bit S-box through various cryptanalysis such as bijective, nonlinearity, linearity, differential cryptanalysis, differential style boomerang attack, avalanche effect, bit independence criterion, etc. Also, a comparison is carried out to exhibit the superiority of the proposed 5-bit S-box over its 5-bit competitors.

1. Introduction

Cryptography is originally from the Greek words, "kryptÓs (hidden/secret) and graphein (to write)", means "secret writing" [1]. It is a technique that converts readable text (known as plain text) into unreadable form (known as a cipher), called encryption and the reverse procedure restores it to its original form, called decryption [2]. It secures the communication by guaranteeing confidentiality, integrity and authentication and authorization of the data [3]. Traditional cryptography could be easily applied to servers, personal computers and smart devices such as smartphones, wearables and other smart gadgets (Fig. 1(A)). But it could not be deployed easily on resource-constrained Internet of Things (IoT) devices such as sensors, RFID tags, actuators, etc., [4] due to their limited memory, small physical area to implement, low computing power and low energy [5]. Such resource limitation challenges could be effectively addressed by its lighter version, called lightweight cryptography [6].

Any cryptography algorithm can be classified into two main categories, symmetric key and asymmetric key cryptography (Fig. 1(B)). Symmetric key cryptography can be classified into three types: block cipher, stream cipher, and hash function. Based on the structure used, block cipher can be further categorized into six subcategories: Substitution-Permutation Network (SPN), Feistel Network (FN), General Feistel Network (GFN), Add-Rotate-XOR (ARX), NonLinear-Feedback Shift Register (NLFSR) and Hybrid [6]. Fig. 1(B) depicts the types of cryptography algorithms, concentrating on the symmetric one. In this work, we have focused on the Substitution technique used in SPN and FN, the two most popular structures in lightweight cryptography [7], by briefing the existing work and by proposing a novel 5-bit substitution box (S-box) that uses enhanced logistic theory [8] for dynamic chaotic behaviour of the elements in the S-box.

Substitution and Permutation are two primitive cryptographic operations introduced by Claude Shannon in 1949 [2]. Substitution is the heart of any SPN based cryptography algorithm. It is achieved through S-box in which each element in the plaintext (bit/letter or group of bits/letters) is mapped into another element to offer confusion property. It makes the relationship as complex as possible between key

* Corresponding author.
E-mail addresses: v.thakor@tees.ac.uk (V.A. Thakor), m.razzaque@tees.ac.uk (M.A. Razzaque), add@eced.svnit.ac.in (A.D. Darji), akshpatel19@gmail.com (A.R. Patel).

https://doi.org/10.1016/j.jisa.2023.103444

Available online 10 February 2023

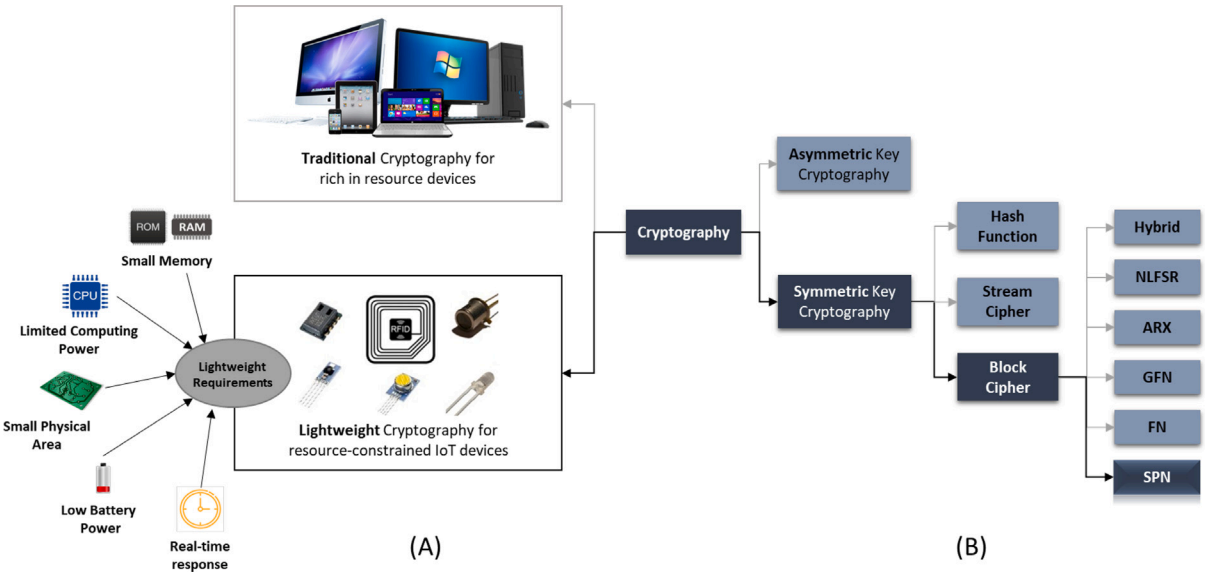


Fig. 1. Classification of cryptography based on (A) Implementation on IoT devices (B) Structure based.

and cipher. On the other hand, permutation rearranges each element of the plain text to offer diffusion property using a permutation table or a technique called transposition. The strength of any cryptography algorithm could be derived from its S-box architecture. Many parameters could define the structure of an S-box, but one of the critical parameters is the input bits it takes. There are various S-boxes with different length input bits such as 3-bit, 4-bits, 5-bit, 6-bit and 8-bits. Generally, the security level goes up as the number of input bits increases, leading to higher resource demand such as memory, physical area, processing power and energy. For instance, 8-bit S-boxes are more secure compared to 4-bit S-boxes, but expensive in terms of resource utilization [6].

The S-boxes are the fundamental elements of symmetric cryptography, providing nonlinearity to the algorithms. In the last decade, a variety of S-boxes have been studied and proposed by various researchers to support different cryptographic applications [9–22]. Some of these are not suitable for lightweight applications due to their heavy structure and high demand for resources (such as 8 and 6 bit S-boxes). In contrast, some suffer during critical cryptanalysis (such as 3 and 4 bit S-boxes). While focusing on the lightweight cryptography algorithms, 4-bit S-boxes are the most popular choice amongst the others due to their compact structure and ease of implementation in [16,20,21,23]. They exhibit excellent performance in resource-constrained environments; however, the security protection is weaker compared to high-end bit S-boxes. The second most popular S-box is 8-bit S-box (variant of AES [24]) due to its robust strength [11,12,17] but requires high amount of resources to get an acceptable performance. Thus, a trade-off amongst performance, cost and security is missing and creates the demand for a balanced S-box.

Recently, many researchers have proposed various S-boxes [9,15,25,26] based on some chaotic theory that shows good resistance against cryptanalysis. However, most of them are 8-bit in size. The comparison of cryptanalysis for these 8-bit S-boxes is showcased, but the performance and cost are not compared with other bit-size S-boxes. Due to their large size (8 × 8 bit), these S-boxes are not suitable for resource-constrained IoT devices or, in other words, the design is not ideal for lightweight cryptography. In addition, very few algorithm designs suit the short messages.

This paper proposes a new 5-bit S-box design that uses the latest chaotic mapping technique suitable for lightweight cryptography algorithms, particularly for small/tiny messages in IoT devices like RFID tags, sensors and smart cards. The article demonstrates the importance of the 5-bit S-box over other n-bit S-boxes by comparing their execution cost. Compared to the resource requirement data available for the 4/5-bit S-box of the popular algorithms, the proposed 5-bit S-box can be implemented using a few resources (area and power). Not only that, but it easily fits with various block lengths (32-bit, 48-bit, 64-bit, 128-bit and 256-bit). And therefore, it could be used easily with different LWC algorithms, particularly with LWC algorithms installed on resource-constrained IoT devices (such as RFID tags, sensors, smart cards, etc.) to play with small/tiny messages. Also, the proposed 5-bit S-box provides significantly better security. The second half of the article demonstrates the security resistance of the proposed 5-bit S-box by comparing it with the same bit-size S-boxes via essential security properties.

Considering the significance of substitution technique in lightweight cryptography algorithms, this article takes an inclusive view on design criteria of S-box to trade-off among performance, cost and security. Section 2 discusses existing n-bit S-boxes along with their advantages and limitations. The proposed 5-bit S-box is discussed in detail in Section 3 by elaborating its design criteria, various schemes to derive over different block sizes and also by demonstrating its performance and implementation cost. One of the crucial trade-offs, security characteristics (cryptanalysis) of the proposed model, is evaluated and documented in Section 4 by comparing it with the same-size existing S-boxes. Finally, Section 5, concludes the proposed work.

2. Existing S-boxes & facts

This section starts with a discussion of existing S-box designs and their advantages and limitations. Further, it reveals the design facts of these existing S-boxes and inspires to development of a new S-box with a balance between cost, performance and security.

2.1. Popular S-boxes

Many researchers and scientists proposed a variety of S-box concepts in the past. Some show high resistance against various attacks and high resource demand, whereas some demonstrate better performance but a weak stand against the security attacks. Most of these S-boxes take 3-bit, 4-bit, 5-bit, 6-bit or 8-bit input and produce either the same or compressed bit output [14]. Among these, 4-bit S-boxes are popular among lightweight cryptography algorithms due to their compact [20,21] but simple implementation [13]. This section presents an overview of S-boxes used by popular lightweight cryptography algorithms such

as PRINT, PRESENT, RECTANGLE, EPCBC, TWINE, LED (Light Encryption Device), SKINNY, Piccolo, KLEIN, Puffin, LBlock, SPONGENT, DESL/DESXL, ASCON, PRIMATE, ICEPOLE, and SHAMASH.

3-bit S-box: PRINT [27], dedicated designed for integrated circuit (IC) printing, offers the smallest 3 × 3-bit S-box, $S : \{0,1\}^3 \rightarrow \{0,1\}^3$ (Table 1). A single S-box in an octal numeral system is used parallelly for $\frac{b}{3}$ times, where $b \in \{48, 96\}$, the size of input block. It is both hardware and software efficient due to its cost-effective implementation on extremely low-cost RFID tags. At the same time, it is vulnerable to attackers due to its small number of possibilities to create different S-boxes.

4-bit S-box: PRESENT [28] uses 4-bit S-box, $S : \{0,1\}^4 \rightarrow \{0,1\}^4$ (Table 2). It relies on of Hexadecimal system and forms a state as sixteen 4-bit words in each sBoxLayer. The S-box design criteria allows 8064 possible S-boxes schemes (maximum) [20]. It is victime of differential cryptanalysis [29,30].

RECTANGLE [31] adopts a 4 × 4 S-box from PRESENT with a reduced number of rounds (25 compared to 31) to offer software efficiency. It has AES like structure with the removal of a few functionalities (a slight change in SP Network) and the introduction of the bit-slice technique to improve performance and cost. Unfortunately, like PRESENT, it also suffers from various cyber-attacks [31].

EPCBC (Electronic Product Code Block Cipher) [32] uses the same 4 × 4 S-box as used in PRESENT. It just varies in key scheduling from PRESENT.

Like the above algorithms, TWINE [33] also use the ready-made 4 × 4 S-box from PRESENT. With other structural changes in the algorithm, it gives faster performance than PRESENT [34].

The trend of using 4 × 4 S-box from PRESENT continues with LED [35], SKINNY [36], Piccolo (four bijective S-boxes) [37], KLEIN [38], Puffin [39], LBlock (such 8 different 4 × 4 bit S-boxes) [40] and SPONGENT (uses it for $\frac{b}{4}$ times parallelly, where b is the fixed number of bits of a state) [41].

5-bit S-box: ASCON [42,43] uses 5-bit S-box, $S : \{0,1\}^5 \rightarrow \{0,1\}^5$, in parallel over 320 bits in bit-slice manner (Table 3). SHAMASH [44], similar to ASCON, uses 5-bit S-box with minor linearity and bit distribution difference compared to ASCON's S-box. PRIMATE [45] works on 5 × 8 and 7 × 8 states of 5-bit elements for multiple times on different variances. ICEPOLE [46,47] operates 5-bit S-box on 256 rows of 1280 state of the plaintext. The structure of all of these S-boxes is remarkably similar. Due to their odd size (not the multiple of two, i.e., $size \neq 2^n$), they are not as popular as 4-bit S-boxes and have limited history.

6-bit S-box: DESL is the lightweight version of DES (Data Encryption Standard), where it is further updated as DESXL with a key whitening feature to improve the security [48]. DESL/DESXL, uses 6-bit S-box that takes 6-bit input and produces compressed 4-bit output [14,48]. Both replaces 8 different 6 × 4 bits S-box of DES with a single 6 × 4 bits S-box, $S : \{0,1\}^6 \rightarrow \{0,1\}^4$. The first and last bits of the input form a 2-bit binary to select one of four rows, and the middle 4-bit selects one of the sixteen columns (Table 4). For instance, 6-bit input 011001, the row is 01 (row 1), and the column is 1100 (column 12) will be selected to produce the output 13 (1101). The possible number of different S-boxes with this design criteria is 2^{56} [48].

8-bit S-box: ICEBERG [49] uses an 8 × 8 S-box, $S : \{0,1\}^8 \rightarrow \{0,1\}^8$ (Table 5) (inspired from AES [24]), spread over 3 stages (S_0, S_1, S_0) in the form of 4 × 4 S-boxes (Tables 6, 7) in parallel to achieve the substitution. Many algorithms use 8-bit S-box, but it is quite expensive in terms of cost and performance while implementing on resource-constrained IoT devices.

Table 1
3-bit S-box design.

x	0	1	2	3	4	5	6	7
$S(x)$	0	1	3	6	7	4	5	2

Table 2
4-bit S-Box Design.

x	0	1	2	3	.	.	.	D	E	F
$S(x)$	C	5	6	A	.	.	.	7	2	9

Table 3
5-bit S-Box Design.

x	0	1	2	3	.	.	.	29	30	31
$S(x)$	4	11	31	20	.	.	.	10	15	23

Table 4
6-bit S-Box Design.

x		0	1	2	.	.	12	13	14	15
$S(x)$	(00)	14	5	7	.	.	0	6	13	3
	(01)	5	0	8	.	.	13	4	1	10
	(10)	4	9	2	.	.	5	11	3	6
	(11)	9	6	15	.	.	0	14	10	13

Table 5
8-bit S-box Design.

	00	01	02	03	.	.	.	0D	0E	0F
00	24	c1	38	e7	.	.	.	d6	52	fd
10	40	6c	d3	3d	.	.	.	fb	fc	f1
.	.	.	.	.	.	.	.	.	.	.
e0	a0	95	65	bf	.	.	.	9b	a4	d1
f0	cb	1f	8f	8e	.	.	.	1e	0f	79

Table 6
S_0: 4 × 4 S-box.

0	1	2	3	.	.	.	B	C	D	E	F
d	7	3	9	.	.	.	5	e	6	0	8

Table 7
S_1: 4 × 4 S-box.

0	1	2	3	.	.	.	B	C	D	E	F
4	a	f	c	.	.	.	6	1	7	3	2

2.2. S-box designs and facts

The facts about S-box observed from the study are as follows:

- 3-bit S-box is the cheapest in terms of memory, energy and computing power along with high performance but can be easily victimized of an attack due to only 2^3 different S-box possibilities.
- 4-bit S-box is more efficient than 8-bit S-box in terms of energy consumption but provides low security (this could be resolved by increasing the number of rounds).
- 5-bit S-box is not widely used due to its odd nature but could be an alternative to 4-bit S-box in terms of improved security.
- 6-bit and 8-bit S-boxes are comparatively more secure than 4-bit S-box but expensive in terms of resources.

Table 8 exhibits the existing n-bit S-boxes and their related concerns.

3. Proposed work

This section gives an inclusive view of the design criteria of the proposed 5-bit S-box for lightweight cryptography algorithms by considering the significance of the substitution technique. Further, it describes

Table 8
Existing S-boxes and related concerns.

S-box type	Facts
3-bit [27]	Implementation cost is very low but could be easily breakable (only 8 possible values)
	Even the increase in the no. of rounds could not help to bring an adequate security level
4-bit [28–31]	Low resource requirements, Low security (only 16 possible values)
	An increase in the no. of rounds could resolve this issue but affects adversely on execution time
5-bit [42–45,47]	A very minor increase in resource requirements compared to 4-bit S-box
	Moderate/adequate security level (32 possible values)
6-bit [48]	Demands little more memory (to store 64 possible values) and little high processing power (to derive/process 64 possible values) compare to 4-bit S-box
	The above demand leads to high energy consumption compared to 4-bit S-box
	The above parameters could increase the demand for the physical area (GE)
8-bit [24,49]	Demands huge memory (to store 256 possible values)and very high processing power (to derive/process 64 possible values)
	The above demand leads to very high energy consumption compared to 4-bit and 6-bit S-box
	The above parameters could dramatically increase the demand for the physical area (GE)

the various schemes to derive over different block sizes using this 5-bit S-box. In addition, cost and performance of the proposed S-box are evaluated by implementing it on the ASIC platform (Application-Specific Integrated Circuit) and compared with its competitor S-boxes.

3.1. Proposed S-box design

The proposed S-box transforms 5-bit of input to a unique 5-bit of output, $S : \{0,1\}^5 \rightarrow \{0,1\}^5 : x \rightarrow S(x)$. The 5-bit S-box consists of 2 rows and 16 columns. The first bit of 5-bit input (i.e., 0 or 1) selects the row, and the remaining four bits decide the column number. The number of columns is equal to half the number of distinct output values in the S-box (2^m, where $m = 5$ is the number of output bits). The 5-bit input creates 2^5 possible input values, and these 2^5 (i.e., 32) values can be easily accommodated into this S-box table.

Fig. 2 demonstrates an example of the proposed 5-bit S-box with randomly placed 32 values (using Enhanced Logistic mapping theory) into a 2×16 table. Further, it demonstrates a unique mapping of 5-bit input into an S-box for matching output. The pseudocode to generate the dynamic chaotic sequence is as follows:

Step 1. Declare decimal constant p
Step 2. Assign p, where $p \geq 2.0$ (In our case, $p = 4.0$)
Step 3. Declare variable v_i
Step 4. Initialized v_i, where $v_i < 1$ (In our case, $v_i = 0.972$)
Step 5. Calculate v_{i+1} using $\sin(\pi p v_i(1 - v_i))$
Step 6. Repeat step 5 for n-times to generate dynamic chaotic sequence (In our case, $n = 32$ times)
Step 7. Finally, arrange the elements $(1, 2, \ldots, n)$ in ascending order of the sequence generated

3.2. Design criteria of 5-bit S-box

To build a simple but robust 5-bit S-box, $S : F_2^5 \rightarrow F_2^5$, that could be easily implemented on resource-constrained IoT device, the following simple but security efficient rules need to apply:

1. S-box, S, must have distinct 32 elements (0–31) spread over 16 columns and 2 rows that satisfies bijective property (Section 4.1).
2. Generate the complex chaotic sequence of the elements in 5-bit S-box using enhanced logistic map equation [8] as defined follows:

$$v_{i+1} = E(L(v_i)) = \sin(\pi p v_i(1 - v_i)) \quad (1)$$

where p is a control parameter, and $p \in (2, +\infty)$. Even though p could have initiated with 0, the suggested initial value of p is 2 for a complex dynamic chaotic behaviour reasons. This is because all the fixed points of the enhanced logistic map are unstable when $p \geq 2$ [9].
3. Any value, V_i, in S, must be different from its column index, C_i, to avoid a fixed point, i.e.,

$$V_i \neq \begin{cases} C_i & \text{if } V_i \in R_0 \\ C_{(15+i)} & \text{if } V_i \in R_1 \end{cases}$$

where, R_0 is the 0th row and R_1 is the 1st row, $R_0 \subset S$ and $R_1 \subset S$
4. An input value, In_i, and its corresponding value, V_i, in S must have bit variation of n bit(s), $0 < n \leq 5$, to meet overall Strict Avalanche Criteria (SAC), i.e.,

$$Bit_{var}(V_i, In_i) \geq n, \quad \forall V_i \in S \quad (2)$$

where In_i: $f(R_i, C_i)$, $R_i \rightarrow \{0,1\}$, $C_i \rightarrow \{0,1\}^4$ and $V_i \rightarrow \{0,1\}^5$

Here, in design criteria (2), the sequence of elements in the 5-bit S-box could be generated using any chaotic mapping methods such as logistic map, sine map, tent map and quadratic map to improve its dynamic behaviour. But, we adopt an enhanced logistic map technique for our 5-bit S-box as it eliminates fixed point [9] weakness of an S-box design.

By implementing the above-defined set of rules, the total number of possible random 5-bit S-box could be $31! \approx 8.22 * 10^{33}$ which is enormous compared to a number of 4-bit S-box that is $15! \approx 1.3 * 10^{12}$.

3.3. Implementation flexibility of 5-bit S-box with different block size

Usually, the input block size are even and of 2^n ($n = 5, 6, 7, \ldots$) [11,16,20]. Also, they are multiple of either 4 or 6, in general. Odd size S-boxes are avoided due to the flexibility to split the input block over the S-box size, and usually 4, 6 or 8 bits are considered.

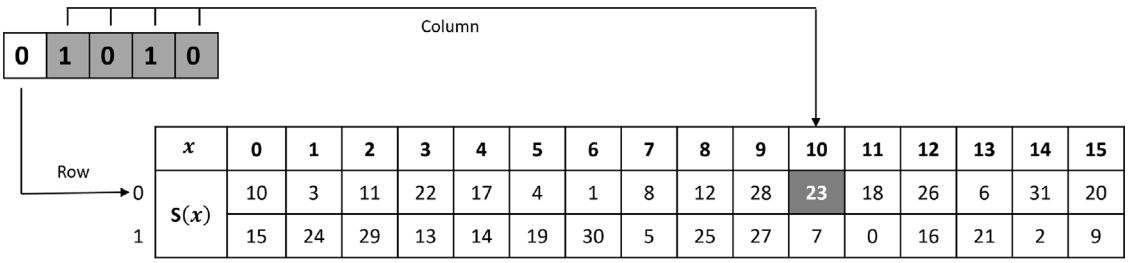
Although the proposed S-box is 5-bit, an odd size S-box, it easily fits over the various input block sizes such as 32, 48, 64, 128 and 256.

Let us consider a 32-bit input block where the middle 30 bits (out of 32) can be split into six 5-bit inputs (to the 5-bit S-box). Then remaining first and last bits can be swapped. Similarly, a n-bit input block can be divided into m 5-bit input (to the 5-bit S-box). Then remaining x (*i.e.*, $n - 5$ m) bits, $x \in \{1, 2, 3, 4\}$, can be interchanged. Table 9 gives the brief of how 5-bit S-box can be implemented with popular input block sizes.

3.4. Performance and cost

We have implemented 5-bit S-boxes on ASIC platform using hardware description language (HDL), Verilog [50], on *Cadence* (Genus) RTL synthesis tool (compiler) using 180 nm SCL180 library to evaluate the cost and performance as shown in Fig. 3.

The estimation of Gate Equivalent (GE) of the logic gates used in the above ASIC implemented can be given as an AND/OR gate costs 0.98 GE, an XOR gate costs 1.96 GE, an XNOR gate costs 2.16 GE, a NOT gate costs 1.18 GE, and a NAND/NOR gate costs 0.78 GE for all 2-input logic gates, whereas a 3-input NAND gate costs 1.37 GE and a 3-input NOR gate costs 0.98 GE.



x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S(x)$	10	3	11	22	17	4	1	8	12	28	**23**	18	26	6	31	20
	15	24	29	13	14	19	30	5	25	27	7	0	16	21	2	9

5-bit Input	Row	Column	Mapping Value	5-bit Output
0 1 0 1 0	(0) 0	(1 0 1 0) 10	23 (1 0 1 1 1)	**1 0 1 1 1**

Fig. 2. Mapping of each 5-bit block into an S-box for its replacement bits.

Table 9
Implementation Flexibility of 5-bit S-box with various block size.

Block size	Implementation on 5-bit S-box	Remaining bits
32-bit	The middle 30 bits (out of 32) can be split into six 5-bit inputs to the 5-bit S-box	The remaining first and last bits can be swapped.
48-bit	The middle 45 bits (out of 48) can be split into nine 5-bit input to the 5-bit S-box	The remaining three bits, either 'first and last two' or 'first two and last' bits, can be interchanged.
64-bit	The middle 60 bits (out of 64) can be split into twelve 5-bit input to the 5-bit S-box	The remaining first two and last two bits can be interchanged.
128-bit	The middle 125 bits (out of 128) can be split into twenty-five 5-bit input to the 5-bit S-box	The remaining three bits, either 'first and last two' or 'first two and last' bits, can be interchanged.
256-bit	The middle 255 bits (out of 256) can be split into fifty-one 5-bit input to the 5-bit S-box	The remaining one bit (either first or last) can be inverted (Ones' complement).

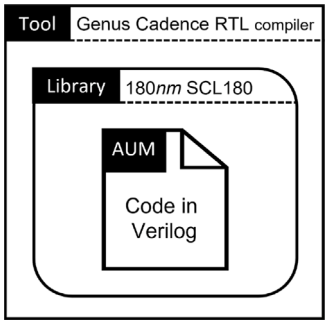


Fig. 3. Experiment setup.

Our S-box consists of eight NAND gates (seven 2-input NAND gates and one 3-input NAND gate) and seven NOR gates (six 2-input NOR gates and one 3-input NOR gate).

Fig. 4 shows the datapath of an area-optimized by 5-bit S-box, which performs one round in one clock cycle, i.e. a 32-bit width datapath at 100 KHz frequency. The experiment witnesses a throughput of 3200 Kbps by consuming around 0.042 μW power to implement this logic.

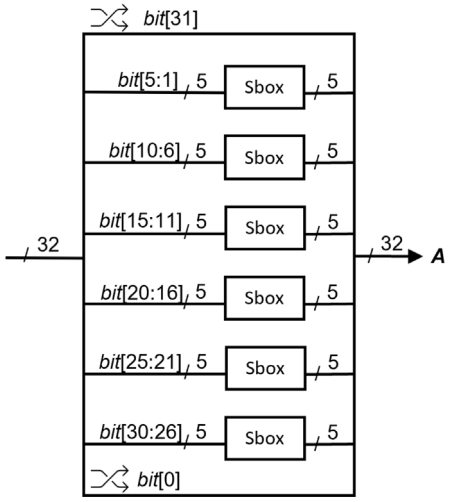


Fig. 4. The datapath of 5-bit S-box.

Table 10
Various S-box area comparison.

Algorithm	Ref	S-box bits	Area (GE)
PRESENT	[28]	4-bit	28.03
	[51]	4-bit	22.67
SKINNY	[36]	4-bit	12–14.68
LED	[35]	4-bit	22.33
Piccolo	[37]	4-bit	24
	[51]	4-bit	12
PRIMATE	[52]	5-bit	30–40
Keccak	[51]	5-bit	17
Proposed	–	**5-bit**	**12.54**

Table 10 compares area requirements (in GE) of the various S-boxes (4-bit/5-bit) used in the popular LWC algorithms. Our proposed 5-bit S-box requires only 12.54 GE and beats out 4-bit/5-bit competitors. This can be visualized in Fig. 5.

4. Security analysis

This section demonstrates the security strength of the proposed 5-bit S-box, measured over bijective property, nonlinearity, linearity (LP), differential probability (DAP), differential style boomerang attack (BCT/FBCT), degree of avalanche effect, bit Independence criteria (BIC) and algebraic attacks. It also gives comparison of cryptanalysis of the proposed 5-bit S-box with other existing 5-bit S-boxes from ASCON [42,43], PRIMATE [45], ICEPOLE [46,47], and SHAMASH [44].

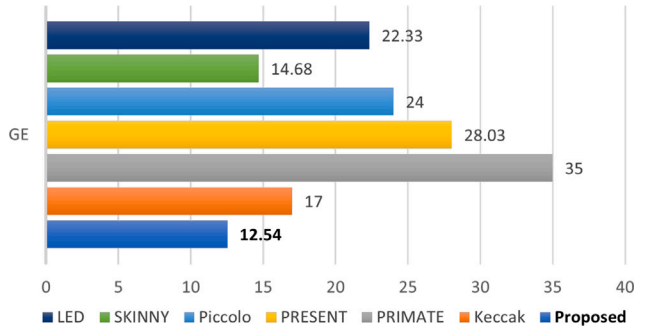


Fig. 5. Various S-box area (GE) comparison.

The superiority of the proposed S-box could be analysed from Table 15 and Fig. 11.

4.1. Bijective property

The bijective property of the proposed m x m S-box (where $m = 5$) is derived using Hamming weight $H_{wt}()$, which is the number of symbols (1) that are different from the zero-symbol (0) of the alphabet (0, 1) used in a string [53], i.e., the number of 1's in the string of bits. It is defined as follows:

$$H_{wt}\left(\sum_{i=1}^{m} b_i f_i\right) = 2^{m-1} \tag{3}$$

where $b_i \in \{0,1\}$ and $(b_1, b_2, \ldots, b_m) \neq (0,0,\ldots,0)$ for each Boolean function, $f_i(1 \leq i \leq m)$. Here, f_i fulfils the bijective property by balancing 0 and 1. Also, the proposed 5-bit S-box has all distinct values from 0 to 31, and thus manifest the bijective property.

4.2. Nonlinearity

S-box operations are designed to obtain nonlinearity to the algorithm. It should not be possible to break the algorithm by solving a set of equations using some set of unknown values. Since the proposed S-box selects its elements at random using a dynamic chaotic mapping system (under a defined set of rules), it is almost impossible to derive an equation that solves any correlations between the input value and the corresponding substitution value. Section 3.2 details how 5-bit input is arbitrary replaced with another 5-bit output. The nonlinearity can be measured either using Hamming distance or the Walsh matrix. Here, the Walsh matrix is not possible to apply as it works on multiple of two, and our proposed S-box makes use of 5 bits. We measure the nonlinearity using Hamming distance (H_d). It is the distance between any corresponding input–output pairs (x_i, y_i), where $H_d(x_i, y_i) = \#(x_i \neq y_i)$. The minimum value of Hamming distance (H_d) for the proposed 5-bit S-box is 1, whereas maximum is 5 (Table 11). The average Hamming distance (H_d) calculated for the proposed 5-bit S-box is 2.625. Fig. 6 depicts the nonlinearity comparison via Hamming distance (H_d) for the proposed S-box and its 5-bit competitors. The higher the Hamming distance (H_d), the higher the nonlinearity property. Thus our proposed 5-bit S-box satisfies the nonlinearity characteristics.

4.3. Linear approximation probability (LP)

Introduced by Matsui's [54], linear approximation probability finds out the maximum value of imbalance in the input–output elements. Let Δx and Δy be the input and output differentials, respectively, and x is a set of all possible inputs with cardinality 2^n. The linear approximation probability for a given S-box is defined as

$$LP = \max_{\Delta x, \Delta y \neq 0} \left| \frac{\#\{x \in X | x.\Delta x = S(x).\Delta y\}}{2^n} - \frac{1}{2} \right| \tag{4}$$

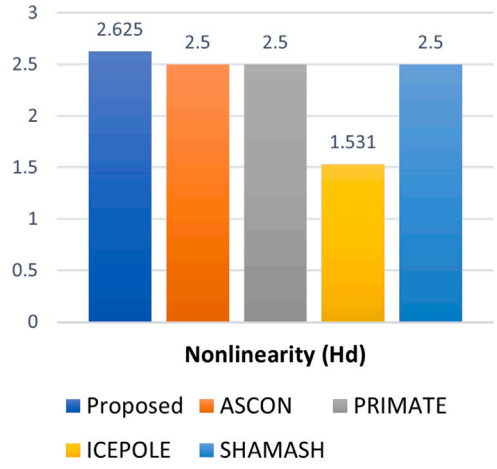


Fig. 6. Nonlinearity (Hamming distance (H_d)).

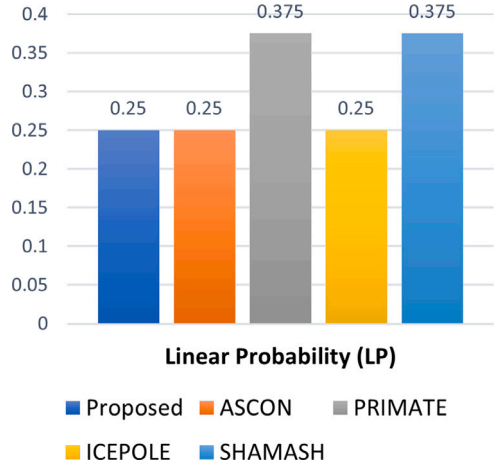


Fig. 7. Linear approximation probability (LP) comparison.

The highest matching event of $x.\Delta x = S(x).\Delta y$, for all $x \in X$, found in the proposed 5-bit S-box is eight, and thus maximum linear approximation probability according to the above equation is 0.25. The value of LP close to zero means better security property. Fig. 7 reveals that the proposed 5-bit S-box has either better or similar linearity property compared to its 5-bit competitors.

4.4. High resistance to Differential Cryptanalysis

Differential Cryptanalysis [55] is a statistical attack using an S-box's Differential Distribution Table (DDT) characteristic. It signifies how the output of an S-box varies as the input is changed. There must be undefined changes in output to protect against Differential Cryptanalysis. It is measured as differential approximation probability (DAP), the differential uniformity of the S-box input–output. It is defined as follows:

$$DAP = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X | S(x) \oplus (S(x \oplus \Delta x) = \Delta y)\}}{2^n} \right) \tag{5}$$

Here, X is the set of all possible input values and n is the number of input bits. DAP is the maximum probability of output difference Δy when the input difference is Δx. For each input value x, $(\Delta x, \Delta y) \in [0, 31]$, the maximum differential approximate probability of the proposed 5-bit S-box is 8, i.e., DAP value is 0.25 ($8/2^5$) (Table 12). Fig. 8 gives the comparison of the differential approximate probability of the various 5-bit S-boxes. For an ideal S-box, the DAP should

Table 11
Nonlinearity measure through Hamming distance (H_d).

Input	Output	Hamming distance (H_d)	Input	Output	Hamming distance (H_d)
0 (00000)	10 (01010)	2	16 (10000)	15 (01111)	5
1 (00001)	3 (00011)	1	17 (10001)	24 (11000)	2
2 (00010)	11 (01011)	2	18 (10010)	29 (11101)	4
3 (00011)	22 (10110)	3	19 (10011)	13 (01101)	4
4 (00100)	17 (10001)	3	20 (10100)	14 (01110)	3
5 (00101)	4 (00100)	1	21 (10101)	19 (10011)	2
6 (00110)	1 (00001)	3	22 (10110)	30 (11110)	1
7 (00111)	8 (01000)	4	23 (10111)	5 (00101)	2
8 (01000)	12 (01100)	1	24 (11000)	25 (11001)	1
9 (01001)	28 (11100)	3	25 (11001)	27 (11011)	1
10 (01010)	23 (10111)	4	26 (11010)	7 (00111)	4
11 (01011)	18 (10010)	3	27 (11011)	0 (00000)	4
12 (01100)	26 (11010)	3	28 (11100)	16 (10000)	2
13 (01101)	6 (00110)	3	29 (11101)	21 (10101)	1
14 (01110)	31 (11111)	2	30 (11110)	2 (00010)	3
15 (01111)	20 (10100)	4	31 (11111)	9 (01001)	3

Table 12
The DAP matrix of the proposed 5-bit S-box.

6	6	6	6	8	6	8	8	8	8	6	4	4	8	8	8
6	6	8	8	8	4	8	6	6	6	6	8	8	4	8	8

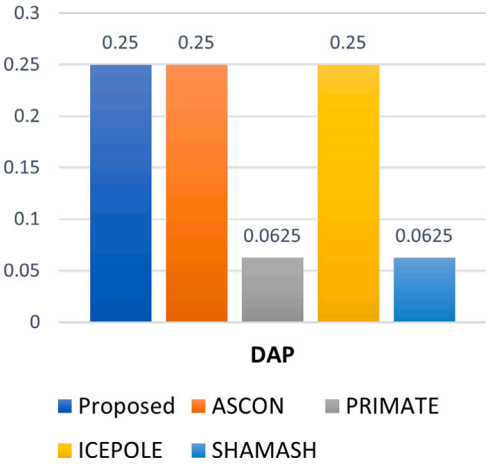


Fig. 8. Differential approximate probability comparison.

be $1/2^n$, which is practically not possible (i.e., it reveals no differential information about the input–output). In other words, the lower the occurrence (DAP), the higher the nonlinearity property. Thus, the proposed S-box shows a good resistance against the differential cryptanalysis even when the size is small (2^5).

4.5. Boomerang Connectivity Table (BCT)

The Boomerang attack [56], proposed by David Wagner, is a differential style attack on block ciphers used to analyse the security of a block cipher. The Boomerang Connectivity Table (BCT) [57] is a systematic approach for calculating the connection probability for a Boomerang attack. Let $S : \{0,1\}^n \rightarrow \{0,1\}^n$ be an invert function, than for a given input difference Δx and output difference Δy for all values of input x, the probability of boomerang of Δx, i.e., BCT of S is given by a 2^n x 2^n table T for all pairs of $(\Delta x, \Delta y)$ as follows:

$$\#\{x \in \{0,1\}^n | S^{-1}(S(x) \oplus \Delta y) \oplus S^{-1}(S(x \oplus \Delta x) \oplus \Delta y) = \Delta x\} \quad (6)$$

Here, S^{-1} is the inverse function of S-box. The values in the boomerang connectivity table are usually greater than or equal to that in the differential distribution table values in terms of strength. This relationship is described in [58]. Table 14 summarizes the occurrence of each element in BCT and DDT of the proposed S-box.

4.6. Feistel counterpart of BCT (FBCT)

The Boomerang Connectivity Table (BCT) is only valid for an S-box that is part of an S-layer in an SPN cipher but not for the S-box that is part of a Feistel cipher. In [59], an extension of BCT is proposed to address the counterpart for a Feistel cipher. Like BCT, Feistel counterpart of BCT (FBCT) for the pairs of $(\Delta x, \Delta y)$ can be given as follows:

$$\#\{x \in \{0,1\}^n | S(x) \oplus S(x \oplus \Delta x) \oplus S(x \oplus \Delta y) \oplus S(x \oplus \Delta x \oplus \Delta y) = 0\} \quad (7)$$

Table 13 exhibits the FBCT of the proposed 5-bit S-box. Here, the values 32, 8, 4 and 0 appear 94, 42, 186 and 702 times, respectively, in the FBCT of the proposed S-box. The highest value in FBCT, known as Feistel boomerang uniformity (β^F), is 8. Here, the FBCT values at the first row, first column, and diagonal is 2^n (i.e., 32) which are neglected. The first row and first column with the values 2^n are known as ladder switch, whereas the diagonal with the values 2^n is known as Feistel switch. Some common properties of any FBCT are as follows [59]:

1. Symmetry: for all $0 \leq \Delta x, \Delta y \leq 2^n - 1$, $\text{FBCT}(\Delta x, \Delta y) = \text{FBCT}(\Delta y, \Delta x)$
2. Fixed values:
   (a) First row: for all $0 \leq \Delta y \leq 2^n - 1$, $\text{FBCT}(0, \Delta y) = 2^n$
   (b) First column: for all $0 \leq \Delta x \leq 2^n - 1$, $\text{FBCT}(\Delta x, 0) = 2^n$
   (c) Diagonal: for all $0 \leq \Delta x \leq 2^n - 1$, $\text{FBCT}(\Delta x, \Delta x) = 2^n$
3. Multiplicity: for all $0 \leq \Delta x, \Delta y \leq 2^n - 1$, $\text{FBCT}(\Delta x, \Delta y) \equiv 0 \bmod 4$
4. Equalities: for all $0 \leq \Delta x, \Delta y \leq 2^n - 1$, $\text{FBCT}(\Delta x, \Delta y) = \text{FBCT}(\Delta x, \Delta x \oplus \Delta y)$

4.7. High degree of avalanche effect

A slight change in input bits that significantly change output bits is known as an avalanche effect. When a change in one input bit results in a change in at least half of the output bits, it is called strict avalanche criterion (SAC)), i.e., for any n bits input, at least $n/2$ bits in output must differ [60]. For any block cipher, an avalanche of change is an essential property and could be boosted by an efficient S-box design that offers high resistance to differential attacks.

As introduced by Webster and Tavares [61], we can confirm whether an S-box fulfil the SAC property or not by considering a 5-bit input X and a set of input vectors, $X_1, X_2, \ldots, X_5$, derived by changing jth bit only. Its corresponding 5-bit output vectors, $Y_1, Y_2, \ldots, Y_5$, can be assigned using a substitution function, $Y_j = S(X_j)$. An avalanche vector,

Table 13
Feistel counterpart of BCT (FBCT) of the proposed 5-bit S-box.

32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
32	32	0	0	0	0	4	4	0	0	0	0	0	0	0	0	4	4	0	0	0	0	8	8	0	0	4	4	0	0	0	0
32	0	32	0	0	0	0	0	0	0	0	0	4	0	4	0	8	8	8	8	0	0	0	0	0	0	0	0	0	4	0	4
32	0	0	32	4	0	0	4	0	4	4	0	0	0	0	0	0	4	4	0	4	4	4	4	4	4	4	4	4	4	4	4
32	0	0	4	32	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	4	0	0
32	0	0	0	0	32	0	0	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	4
32	4	0	0	0	0	32	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	0	0	0
32	4	0	4	4	0	4	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	4	0	0
32	0	0	0	0	4	0	0	32	0	0	0	0	4	0	0	0	4	4	0	4	4	0	0	0	4	4	0	4	4	0	0
32	0	0	4	0	0	0	0	0	32	4	0	0	0	0	0	0	4	8	0	0	0	0	0	4	0	0	8	0	0	0	0
32	0	0	4	0	0	0	0	0	4	32	0	0	0	0	0	0	4	8	0	0	0	0	0	8	0	0	4	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	8	0	0	0	0	0	0	8	0	0	0	0	0	0
32	0	4	0	0	0	0	0	0	0	0	0	32	0	4	0	0	0	0	0	0	0	4	0	0	0	4	0	0	0	0	0
32	0	0	0	0	4	0	0	4	0	0	0	0	32	0	0	0	0	0	4	0	0	0	0	4	0	0	0	0	0	0	0
32	0	4	0	0	0	0	0	0	0	0	0	4	0	32	0	0	0	0	4	4	0	4	0	4	0	4	0	0	4	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	4	4	0	0	4	4	0	0	0	0	0
32	4	8	0	0	0	0	0	0	0	0	0	0	0	0	0	32	4	8	0	0	0	0	0	0	0	0	0	0	0	0	0
32	4	8	4	0	0	0	0	4	4	4	0	0	0	0	0	4	32	4	8	0	0	0	0	0	4	4	0	4	0	0	0
32	0	8	4	0	0	0	0	4	8	8	8	0	0	0	0	8	4	32	0	0	0	0	0	8	8	4	8	0	0	0	0
32	0	8	0	0	0	0	0	0	0	0	0	0	0	4	0	0	8	0	32	0	0	0	0	0	0	0	0	0	4	0	0
32	0	0	4	0	0	0	0	4	0	0	0	0	4	4	0	0	0	0	0	32	0	0	4	0	4	4	0	4	0	0	0
32	0	0	4	0	0	0	0	4	0	0	0	0	0	0	4	0	0	0	0	0	32	4	0	0	0	4	0	0	4	0	0
32	8	0	4	0	0	0	0	0	0	0	0	4	0	4	4	0	0	0	0	0	0	4	32	8	4	4	4	0	0	0	0
32	8	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	8	32	0	0	0	0	0	0	0	0
32	0	0	4	0	0	0	0	0	4	8	0	0	0	4	0	0	4	8	0	0	0	4	0	32	0	0	4	0	0	0	0
32	0	0	4	4	0	0	0	4	0	0	8	0	4	0	4	0	4	8	0	4	0	4	0	0	32	4	0	0	4	0	0
32	4	0	4	0	4	8	4	4	0	0	0	4	0	4	4	0	0	4	0	4	4	4	0	0	4	32	4	8	4	0	4
32	4	0	4	0	0	0	0	0	8	4	0	0	0	0	0	0	4	8	0	0	0	0	0	4	0	4	32	0	0	0	0
32	0	0	4	0	0	8	0	4	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	8	0	32	0	0	4
32	0	4	4	4	0	0	4	4	0	0	0	0	0	4	0	0	0	0	4	0	4	0	0	0	4	4	0	0	32	4	4
32	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	32	0
32	0	4	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	4	0	32

Table 14
The occurrence of each element in BCT and DDT of the proposed S-box.

	32	16	14	12	10	8	6	4	2	0
BCT	63	2	1	8	9	30	72	178	228	433
DDT	1	–	–	–	–	4	13	72	297	637

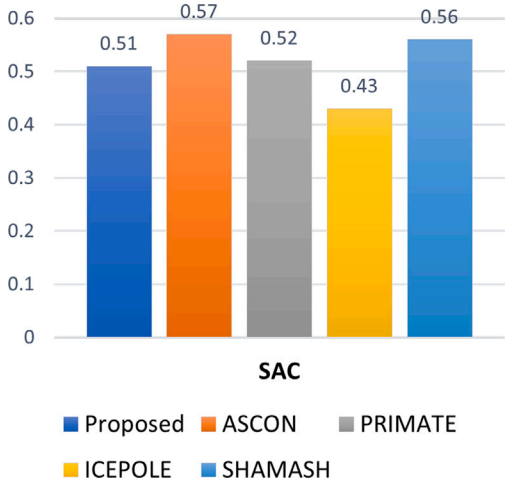


Fig. 9. Avalanche effect/Strict Avalanche Criterion (SAC).

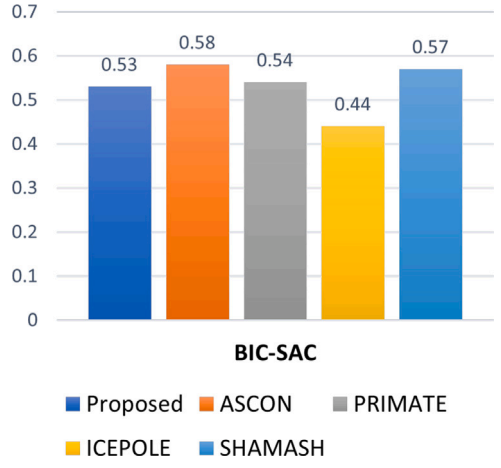


Fig. 10. Bit Independence Criterion — SAC.

V_j, can be computed by XORing output vector Y and Y_j, i.e., $V_j = Y \oplus Y_j$. A 5×5 dependency matrix, A, can be calculated by adding ith bit of V_j to $a_{i,j}$, where $a_{i,j}$ is the ith element of the matrix A. Now, depending on the vector X, repeat the above steps multiple times and then divide each element of matrix A by 2^n (n is the number of input/output bits) to compute SAC matrix.

The average avalanche of change effect of the proposed algorithm is 0.51 (51.00%) (Table 15). If the average of each element and matrix is close the ideal value, 0.5 (50%), then the S-box is considered to satisfy SAC. Thus, the proposed S-box fulfils the SAC property. Fig. 9 compares the avalanche of change effect (SAC) of the proposed S-box and other existing 5-bit S-boxes. It can be observed that the average SAC value of the proposed S-box is closest to the ideal value (0.5), and thus it beats the competition.

Since SAC works on the concepts of completeness along with the avalanche effect, it can be proven below. Let us consider a multi-output function $S : F_2^n \to F_2^m$ which generates m-bit output in responds to n-bit input. Let X be an input where $X = (x_1, x_2, \ldots, x_n)$, $x_p \in \{0, 1\}$ and X^i be another input with change in ith bit(s) where $i = 1, 2, \ldots, n$. Let $F(X)$ and $F(X^i)$ be the corresponding outputs, $F \in T$ where $T \subset S_2^5$.

Definition 1. For multi-output function, $S : F_2^n \to F_2^m$, each output bit relies on every input bit, called *Completeness*, as follows:

$$C = 1 - \frac{1}{nm} \#\{(i, j) | b_{ij} = 0\} \tag{8}$$

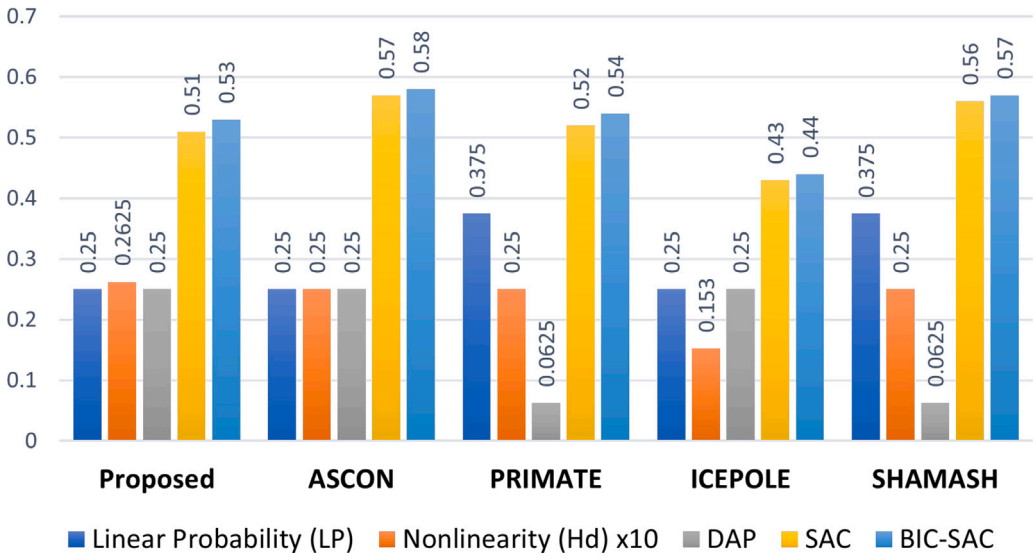


Fig. 11. Cryptanalysis of various 5-bit S-boxes.

where $b_{ij} = \#\{X \in T | (f(X))_j \neq (f(X^i))_j\}$, is the total of resulting output bits when two inputs with ith bit difference is passed, also $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The value of completeness, C, closer to 1 offers strong non-linearity. Our S-box satisfies this property ($C = 1$) as it generates unique pairs of input–output using 5-bits.

Definition 2. For any function, F, if a change in an input reflects the change in half of the output, then it presents a strict avalanche property (SAC) as follows:

$$Avl\_eft_{(strict)} = 1 - \frac{2}{\#T * nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |b_{ij} - \frac{1}{2}\#T \tag{9}$$

Here, $\#T$ is the number of inputs. Similar to *completeness*, the $Avl\_eft_{(strict)}$ value close to 1 shows high avalanche effect. Our S-box shows strict avalanche criterion (SAC) value 0.51 and thus $Avl\_eft_{(strict)} \approx 1$.

4.8. Bit independence criterion

Another essential property, *bit independence criterion (BIC)* introduced by Webster and Tavares [61], where each input bit affects/changes every output bit, i.e., a change in ith bit reflects an independent change of output bits j and k, where $i, j, k \in (1, 2, \ldots, n)$ and $j \neq k$. According to this, two output bits of an S-box, f_j and f_k, where $j \neq k$, if $f_j \oplus f_k$ shows high nonlinearity and fulfil the SAC, the S-box has good BIC property (Fig. 10).

BIC-SAC property can be computed by determining output vectors $Y_1, Y_2, \ldots, Y_5$ for each input vector X as defined in the previous Section 4.7. An avalanche vector, $V_{i,j,k}$, can be computed by XORing $P_{i,j}$ and $Q_{i,j}$, i.e., $V_{i,j,k} = P_{i,j} \oplus Q_{i,j}$. Here, $P_{i,j}$ is the XORed value of ith and jth bit of Y and $Q_{i,j}$ is the XORed value of ith and jth bit of Y_k, where $i, j, k \in \{1, 2, \ldots, 5\}$. Now, depending on the vector X, repeat the above steps multiple times and then divide each element of matrix A by 2^n (n is the number of input/output bits) to obtain BIC-SAC matrix.

To better the resistance property, an S-box must show the BIC-SAC value close to 0.5. The BIC-SAC value for the proposed S-box is 0.53, and thus it satisfies the BIC-SAC property. Fig. 10 illustrates a comparison of BIC-SAC values of the proposed S-box and other 5-bit S-boxes competitors where the average BIC-SAC value of the proposed S-box is closest to the ideal value (0.5), and thus it wins the race.

Table 15
Cryptanalysis of various 5-bit S-boxes.

S-box (5-bit)	Linear Probability	Nonlinearity (H_d)	DAP	SAC	BIC-SAC
Proposed	**0.25**	**2.625**	**0.25**	**0.51**	**0.53**
ASCON	0.25	2.5	0.25	0.57	0.58
PRIMATE	0.375	2.5	0.0625	0.52	0.54
ICEPOLE	0.25	1.531	0.25	0.43	0.44
SHAMASH	0.375	2.5	0.0625	0.56	0.57

4.9. Algebraic attacks

The proposed S-box has a simple but robust structure. Based on the design criteria we proposed for the 5-bit S-box, as discussed in Section 3.2, the possible S-boxes are $31! \approx 8.22 * 10^{33}$ which is huge and noticeably more than that of the 4-bit S-box, i.e., $15! \approx 1.3 * 10^{12}$. Also, the proposed 5-bit S-box make use of a complex dynamic chaotic system to create randomness of the element in the S-box, and it is tough to breakthrough. Moreover, the results achieved through the S-box Evaluation Toolbox (SET) [62], the algebraic immunity of the proposed S-box is 2, which is excellent and similar to its 5-bit S-box competitors.

5. Conclusion

While S-box is the fundamental and the only component that offers a nonlinear functionality in any SPN-based cryptography algorithm, its design significantly impacts its cost, performance and security features. Various researchers have proposed different S-boxes with different bit lengths (e.g., 4/6/8-bit), but the 4-bit S-box from PRESENT is widely used due to its low resource requirements in constrained environments. Also, the 8-bit S-box attracts designers due to its promising security structure, but it witnesses high implementation costs. The other S-boxes, 3-bit and 6-bit, are far from the competition because of either low-security support or expensive implementation reasons. However, the 5-bit S-box structure has limitations in going with popular block sizes; the proposed new S-box design solves this issue by easing flexibility to fit various block sizes with low resource requirements.

The experimental results demonstrate that even if the S-box size is increased by one bit (proposed 5-bit S-box), the resource requirement does not really increase, particularly in the area. But, it boosts the security level significantly and encourages using 5-bit S-box over a 4-bit S-box. The comparison could be extended in the future by adding more

metrics, such as throughput and power/energy consumption, which are not delicately available for the S-boxes (only available for the full algorithm) we compared. Also, a cryptanalysis comparison of the proposed 5-bit S-box with the same bit-size S-boxes over the essential security parameters exhibits the superiority of the proposed S-box.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Rivest RL. Cryptography. In: Algorithms and complexity. Elsevier; 1990, p. 717–55.
[2] Stallings W, Tahiliani MP. Cryptography and network security: principles and practice, vol. 6. Pearson London; 2014.
[3] Mohd BJ, Hayajneh T. Lightweight block ciphers for IoT: Energy optimization and survivability techniques. IEEE Access 2018;6:35966–78.
[4] McKay K, Bassham L, Turan MS, Mouha N. Report on lightweight cryptography (NISTIR8114). National Institute of Standards and Technology (NIST); 2017.
[5] Mohd BJ, Hayajneh T, Vasilakos AV. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. J Netw Comput Appl 2015;58:73–93.
[6] Thakor VA, Razzaque MA, Khandaker MR. Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. IEEE Access 2021.
[7] Hatzivasilis G, Fysarakis K, Papaefstathiou I, Manifavas C. A review of lightweight block ciphers. J Cryptogr Eng 2018;8(2):141–84.
[8] Hua Z, Zhou B, Zhou Y. Sine chaotification model for enhancing chaos and its hardware implementation. IEEE Trans Ind Electron 2018;66(2):1273–84.
[9] Hua Z, Li J, Chen Y, Yi S. Design and application of an S-box using complete latin square. Nonlinear Dynam 2021;104(1):807–25.
[10] Easttom W. S-box design. In: Modern cryptography. Springer; 2021, p. 187–204.
[11] Alshammari BM, Guesmi R, Guesmi T, Alsaif H, Alzamil A. Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic S-box. Symmetry 2021;13(1):129.
[12] Siddiqui N, Yousaf F, Murtaza F, Ehatisham-ul Haq M, Ashraf MU, Alghamdi AM, et al. A highly nonlinear substitution-box (S-box) design using action of modular group on a projective line over a finite field. Plos One 2020;15(11):e0241890.
[13] Dey S, Ghosh R. 4, 8, 32, 64 Bit substitution box generation using irreducible or reducible polynomials over Galois field GF (P q) for smart applications. In: Security in smart cities: models, applications, and challenges. Springer; 2019, p. 279–95.
[14] De Meyer L, Vaudenay S. DES S-box generator. Cryptologia 2017;41(2):153–71.
[15] Yi L, Tong X, Wang Z, Zhang M, Zhu H, Liu J. A novel block encryption algorithm based on chaotic S-box for wireless sensor network. IEEE Access 2019;7:53079–90.
[16] Prathiba A, Bhaaskaran V. Lightweight S-box architecture for secure internet of things. Information 2018;9(1):13.
[17] Lu Q, Zhu C, Wang G. A novel S-box design algorithm based on a new compound chaotic system. Entropy 2019;21(10):1004.
[18] Lambić D. A novel method of S-box design based on discrete chaotic map. Nonlinear Dynam 2017;87(4):2407–13.
[19] Farwa S, Shah T, Idrees L. A highly nonlinear S-box based on a fractional linear transformation. SpringerPlus 2016;5(1):1–12.
[20] Zhang W, Bao Z, Rijmen V, Liu M. A new classification of 4-bit optimal S-boxes and its application to Present, Rectangle and Spongent. In: International workshop on fast software encryption. Springer; 2015, p. 494–515.
[21] Saarinen M-JO. Cryptographic analysis of all 4× 4-bit S-boxes. In: International workshop on selected areas in cryptography. Springer; 2011, p. 118–33.
[22] Lineham A, Gulliver TA. Heuristic S-box design. Contem Eng Sci 2008;1(4):147–68.
[23] Meyer LD. Looking at the NIST lightweight candidates from a masking point-of-view. 2020, Cryptology ePrint Archive, Report 2020/699 https://ia.cr/2020/699.
[24] Daemen J, Rijmen V. AES proposal: Rijndael. MD, USA: Gaithersburg; 1999.
[25] Hua Z, Zhu Z, Chen Y, Li Y. Color image encryption using orthogonal latin squares and a new 2D chaotic system. Nonlinear Dynam 2021;104(4):4505–22.
[26] Si Y, Liu H, Chen Y. Constructing keyed strong S-box using an enhanced quadratic map. Int J Bifurcation Chaos 2021;31(10):2150146.
[27] Knudsen L, Leander G, Poschmann A, Robshaw MJ. PRINTcipher: A block cipher for IC-printing. In: International workshop on cryptographic hardware and embedded systems. Springer; 2010, p. 16–32.
[28] Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJ, et al. PRESENT: An ultra-lightweight block cipher. In: International workshop on cryptographic hardware and embedded systems. Springer; 2007, p. 450–66.
[29] Blondeau C, Nyberg K. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In: Annual international conference on the theory and applications of cryptographic techniques. Springer; 2014, p. 165–82.
[30] Jeong K, Lee Y, Sung J, Hong S. Improved differential fault analysis on PRESENT-80/128. Int J Comput Math 2013;90(12):2553–63.
[31] Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I. RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. Sci China Inf Sci 2015;58(12):1–15.
[32] Yap H, Khoo K, Poschmann A, Henricksen M. EPCBC-a block cipher suitable for electronic product code encryption. In: International conference on cryptology and network security. Springer; 2011, p. 76–97.
[33] Suzaki T, Minematsu K, Morioka S, Kobayashi E. Twine: A lightweight, versatile block cipher. In: ECRYPT workshop on lightweight cryptography, Vol. 2011. 2011.
[34] Toshihiko O. Lightweight cryptography applicable to various iot devices. NEC Tech J 2017;12(1):67–71.
[35] Guo J, Peyrin T, Poschmann A, Robshaw M. The LED block Cipher. In: International workshop on cryptographic hardware and embedded systems. Springer; 2011, p. 326–41.
[36] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Annual international cryptology conference. Springer; 2016, p. 123–53.
[37] Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, Shirai T. Piccolo: An ultra-lightweight blockcipher. In: International workshop on cryptographic hardware and embedded systems. Springer; 2011, p. 342–57.
[38] Gong Z, Nikova S, Law YW. KLEIN: A new family of lightweight block Ciphers. In: International workshop on radio frequency identification: security and privacy issues. Springer; 2011, p. 1–18.
[39] Cheng H, Heys HM, Wang C. Puffin: A novel compact block cipher targeted to embedded digital systems. In: 2008 11th EUROMICRO conference on digital system design architectures, methods and tools. IEEE; 2008, p. 383–90.
[40] Wu W, Zhang L. LBlock: A lightweight block cipher. In: International conference on applied cryptography and network security. Springer; 2011, p. 327–44.
[41] Bogdanov A, Knežević M, Leander G, Toz D, Varıcı K, Verbauwhede I. SPONGENT: A lightweight hash function. In: International workshop on cryptographic hardware and embedded systems. Springer; 2011, p. 312–25.
[42] Dobraunig C, Eichlseder M, Mendel F, Schläffer M. Ascon v1. 2. submission to the CAESAR competition. Inst Appl Inf Proc Commun Graz 2016.
[43] Dobraunig C, Eichlseder M, Mendel F, Schläffer M. Ascon v1. 2: Lightweight authenticated encryption and hashing. J Cryptol 2021;34(3):1–42.
[44] Penazzi D, Montes M. Shamash (and shamashash)(version 1). Lightweight Cryptogr Standard Proc Round 2019;1.
[45] Andreeva E, Bilgin B, Bogdanov A, Luykx A, Mendel F, Mennink B, et al. PRIMATEs v1. 2014, Submission to CAESAR.
[46] Morawiecki P, Gaj K, Homsirikamol E, Matusiewicz K, Pieprzyk J, Rogawski M, et al. ICEPOLE: High-speed, hardware-oriented authenticated encryption. In: International workshop on cryptographic hardware and embedded systems. Springer; 2014, p. 392–413.
[47] l Morawiecki P, Gaj K, Homsirikamol E, Matusiewicz K, Pieprzyk J, Rogawski M, et al. ICEPOLE v2. 2015, CAESAR Submission: http://competitions.cr.yp.to/round2/icepolev2.Pdf.
[48] Poschmann A, Leander G, Schramm K, Paar C. New light-weight crypto algorithms for RFID. In: 2007 IEEE international symposium on circuits and systems. IEEE; 2007, p. 1843–6.
[49] Standaert F-X, Piret G, Rouvroy G, Quisquater J-J, Legat J-D. ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware. In: International workshop on fast software encryption. Springer; 2004, p. 279–98.
[50] Verilog. Wikimedia Foundation; 2021, URL https://en.wikipedia.org/wiki/Verilog.
[51] Picek S, Mariot L, Yang B, Jakobovic D, Mentens N. Design of S-boxes defined with cellular automata rules. In: Proceedings of the computing frontiers conference. 2017, p. 409–14.
[52] Šijačić D, Kidmose AB, Yang B, Banik S, Bilgin B, Bogdanov A, et al. Hold your breath, PRIMATEs are lightweight. In: International conference on selected areas in cryptography. Springer; 2016, p. 197–216.
[53] Wei VK, Yang K. On the generalized hamming weights of product codes. IEEE Trans Inform Theory 1993;39(5):1709–13.
[54] Matsui M. Linear cryptanalysis method for DES cipher. In: Workshop on the theory and application of of cryptographic techniques. Springer; 1993, p. 386–97.
[55] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol 1991;4(1):3–72.
[56] Wagner D. The boomerang attack. In: International workshop on fast software encryption. Springer; 1999, p. 156–70.

[57] Cid C, Huang T, Peyrin T, Sasaki Y, Song L. Boomerang connectivity table: A new cryptanalysis tool. In: Annual international conference on the theory and applications of cryptographic techniques. Springer; 2018, p. 683–714.
[58] Song L, Qin X, Hu L. Boomerang connectivity table revisited. application to SKINNY and AES. IACR Trans Symmetr Cryptol 2019;118–41.
[59] Boukerrou H, Huynh P, Lallemand V, Mandal B, Minier M. On the feistel counterpart of the boomerang connectivity table. IACR Trans Symmetr Cryptol 2020;2020(1):331–62.
[60] Feistel H. Cryptography and computer privacy. Sci Am 1973;228(5):15–23.
[61] Williams H, Webster A, Tavares S. On the design of s-boxes. In: Advances in cryptology—CRYPTO'85 proceedings, Vol. 218. 1986, p. 523–34.
[62] Picek S, Batina L, Jakobović D, Ege B, Golub M. S-box, SET, Match: A Toolbox for S-box Analysis. In: Naccache D, Sauveron D, editors. 8th IFIP international workshop on information security theory and practice. WISTP, Information security theory and practice. securing the internet of things, vol.LNCS-8501, Heraklion, Crete, Greece: Springer; 2014, p. 140–9. http://dx.doi.org/10.1007/978-3-662-43826-8_10, URL https://hal.inria.fr/hal-01400936.