Design of Compact S Box for Resource Constrained Applications

R.Sherine Jenny, R.Sudhakar, M.Karthikpriya

Department of Electronics and communication Engineering,
Dr Mahalingam College of Engineering and Technology

sjenny98@gmail.com

Abstract. The era of IoT has brought huge impact in the lives of people. It changed the way the people interact with the outside world and led to the development of many smart devices. The important factor to be considered in smart devices is security as the sensor nodes are prone to vulnerable attacks. To incorporate security in IoT lightweight block ciphers are preferred. Lightweight block ciphers can be tuned according to applications. This can be motivated by optimizing the S-box since S-box has significant impact on area. Reducing the area produces an impact on power, delay and memory. Here area optimized S-box of lightweight ciphers that serves for IoT devices is proposed. The S-box of PRESENT, GIFT and LED are optimized using karnaugh mapping. Further factorization is carried out to utilize the redundant terms which in turn reduces the gate count. The proposed method is found to be optimum when compared with previous works and it also consumes less area and power.

1. Introduction

The era of Internet-of-Things (IoT) has given rise to a number of smart devices with the ability to communicate with each other across heterogeneous network interfaces. Wireless sensor networks (WSNs) and RFID technology seen a vast growth in it due to the implementation of IoT. IoT devices are widely used in industrial applications, medical monitoring, home automation, and traffic surveillance. Internet of things (IoT) collects information through wireless sensor network. Wireless sensor network (WSN) composes of sensor nodes/devices. Sensor node has the inbuilt capability of sensing, processing and communication. These sensing devices are prone to attacks and bring in many security challenges. Emerging IoT devices targets energy, efficiency, cost, noise and performance. IoT devices require limited battery life and significant physical constraints [1]. IoT devices are quite vulnerable to attacks since they are more accessible to an attacker. The limitations and the cost constraints make the IoT devices quite challenging [2]. There are many threats to privacy when dealing with IoT. At present, IoT is recognized in families, workplaces, social facilities, business companies, etc. who face security and privacy issues. People wish everything to be done in online mode. Since everything becomes online, data thefts can occur very easily and there comes threat to privacy. Let us consider a scenario, in which people use their smart cards to drag amount from their account without even going for bank. This becomes quite easier for the people because just a swipe is enough. In this case privacy and security are major concerns for people. Obviously there is an emerging need of security algorithms for IoT. IoT devices need to perform certain level of secure computation. This is where the cryptography comes in. Many cryptographic algorithms are being developed to enhance the security. The traditional cryptographic algorithms are huge in terms of size, power consumption and...
memory usage which make it unsuitable to be implemented in IoT applications [3]. Further algorithm such as Advanced Encryption Standard (AES) is not suitable for IoT devices due to the computationally expensive mathematical operations. This led to the development of lightweight block ciphers with compact implementation of non-linear S-box for IoT devices. Lightweight block ciphers are in immense demand for Internet of Things applications as they need less resource and have small hardware footprints. Lightweight cryptography is essential in order to protect the IoT devices from various attacks[3]-[4]. Since the secret key stored in an unprotected AES engine can be obtained by attackers by performing correlation power analysis (CPA) attacks, by exploiting certain physical leakage channels etc…[5].

Lightweight cryptographic primitives are preferred to provide security in IoT devices. These lightweight cryptographic algorithms makes use of 4×4 S-boxes. Security –performance trade off depends on the selection of the S-box in the lightweight block ciphers. The choice of the 4×4 S-box for the lightweight constructions results in compact hardware, unlike the 8×8 S-box used in the AES. A large number of the lightweight ciphers namely PRESENT, RECTANGLE, KLEIN, GIFT, SERPENT, NOEKEON, PRINT and PRINCE, have the 4×4 S-box in their structure [6]. Here we have considered some of these lightweight ciphers in our work. Since area and cost becomes major concern for IoT devices, research works were carried out in the past to reduce the size and cost of lightweight block ciphers. Our proposed work reduces the size and cost of some of the lightweight block ciphers using Boolean S-box instead of LUT. Karnaugh mapping is used to implement the S-box structure. Further factorization is used to minimize the size of the S-box.

This paper is categorized as follows: Section 2 gives description about S-box and its importance. Section 3 describes the related works on AES S-box. Section 4 provides a description on lightweight block ciphers as a solution for resource constrained devices with some related works. Section 5 illustrates the work proposed in compact S-box. Section 6 presents the discussion, results obtained and conclusion.

2. S-box and its importance

Substitution box is considered to be the important step in any cryptographic algorithm. Besides the size of S-box, its properties makes it to be the best part in any cryptographic algorithm. Kamsiah Mohamed et al [7] conducted a study on S-box properties to uncover the strength of S-box. Its strength was found to lie in various properties such as Robustness, Balancing, Strict Avalanche Criterion (SAC), Nonlinearity, Differential uniformity, Linear approximation, Algebraic complexity, Fixed and opposite fixed points and Bit avalanche criterion. S-box can be implemented using different methods. Abhishek kumar et al [8] made an analysis to find the best method for implementing S-box. S-box was implemented using various methods such as Look up table or ROM based, Modified lookup table based approach, Computational method, Blend of computation method and LUT method. It was found that LUT based method increased area and delay whereas GF method increased power consumption. S-box was implemented using various methods to increase its strength. A novel approach for S-box generation algorithm was proposed by UnalKavusoglu et al [9]. It was based on chaotic scaled zhongtang system and was created using RNG. Another design for strong S-box was proposed by J.A. Aboytes-Gonzáleet al [10] and it was based on matrix approach. On comparison with older S-boxes the proposed S-box was found to be versatile and efficient to implement. SajjadShaukat Jamalet al [11] proposed S-box using linear fractional transformation and enhanced chaos. It was found that the increase in chaotic range provides best results for secure communication.

3. AES S-box

Advanced Encryption standard is the most widely used cryptographic algorithm. Since S-box is the only non-linear portion in AES many researches were made with respect to this area. On reviewing the past researches, S-box was found to be designed using various techniques to improve its efficiency. Different techniques were proposed to develop a compact S-box suitable for variety of applications.

Xiaoqiang ZHAN et al [12] proposed a compact composite field S-box based on AND-XOR array structure. Multiplication and multiplication inverse is used to reduce the computational complexities
by constructing AND-XOR structure. Common sub expression algorithm (CSE) is used to reduce the redundant gates in the algorithm. This method of AES S-box achieves shortest CPD with smaller area and cost. Nabihah Ahmadet al [13] proposed a full custom CMOS design of Compact composite field AES S-box. They used Pass Transistor logic and analysed to measure the area, delay and throughput. This approach minimized the silicon area, critical path propagation delay and power dissipation. An optimized S-box architecture for Low Power AES Design using multistage PPRM was proposed by Sumiomorioko et al [14]. This method of S-box consumes low power of about 29 µW at 10 MHz when implemented using CMOS technology. A new composite field AES S-box architecture was derived by M. M. Wonget al [15] that achieves an optimally balanced construction in terms of area of implementation and critical path. This architecture achieves high throughput of about 3.49 Gbps in FPGA implementation.

4. Lightweight Block Ciphers

As the IoT started to rule the world many researches were made to secure IoT devices. Because the IoT devices are vulnerable to cyber-attacks. The search of optimum ciphers to secure the resource constrained devices lead to the development of lightweight block ciphers. In traditional block ciphers the S-box is designed as 8 x 8 whereas for resource constrained devices it has been reduced to 4 x 4 and these types of ciphers were also developed. These are intentionally developed for small hardware footprint devices such as RFID, IoT etc...Substitution box is the important portion in any lightweight block ciphers as it is well known for its non-linearity. The general concept of Lightweight block ciphers is shown in Fig. 1.

Some research works before the development of lightweight block ciphers are given as follows. Prasetyo et al [16] proposed data encryption for IoT using BLOWFISH algorithm by reducing the number of rounds. Xuanxia Yao et al [17] proposed lightweight no-pairing elliptic curve cryptography (ECC) based on attribute-based encryption (ABE) to address security and privacy issues in IoT. Though the works were not satisfied, the research of ciphers for IoT devices continued. This lead to the successful development of lightweight block ciphers by Andrey Bogdanov. Later Poonam Jindalet al [3]made an intensive study on lightweight cryptography as a solution to secure IoT. 21 lightweight block ciphers, 19 lightweight stream ciphers, 9 hash functions and 5 variants of elliptic curve cryptography were studied. Area minimized circuits for 16-bit S-boxes, procedure to identify cryptographically significant S-box constructions based on power mappings over GF (2^n) and searching for composite-field representations that permit low-area hardware implementations was proposed by Christopher A. Woodet al [18]. A.Prathibaet al [4]designed lightweight S-box architecture for secure Internet of Things where A combinational non linear 4 x 4 S-box was designed and devised in finite fields. It was observed that the sub field expressions of GF ((2^2)^2) requires 86.5% lesser number of gates when compared to GF (2^4). After the development of lightweight block ciphers, A.Prathibaet al [19]proposed hardware footprint of S-box in lightweight block cipher for IoT and CPS information security systems. This was made in order to reduce the power-delay product, area-power product and area-delay product. J.J.Tay[20] proposed Boolean S-box of PRESENT with reduced number of gates when compared to the original S-box. Mohamad Sbeiti[21] implemented PRESENT in FPGA board to investigate the performance of PRESENT. The results showed that PRESENT is efficient in terms of cost, area and hardware efficiency when compared to AES, ICEBERG and SEA algorithms. The hardware implementation of PRESENT protected against SCA attacks was proposed by Xiao Yu[22]. Since S-box consumes more space, reduction of size has become an ongoing research area. The Lightweight block ciphers considered in this paper are PRESENT, LED, GIFT, KLEIN and RECTANGLE.
4.1 PRESENT
PRESENT is symmetric encryption algorithm. It was specifically designed with ultra-constrained applications such as passive low-cost RFID-tags in mind [13]. PRESENT is a so-called substitution-permutation network (SPN) with a block size of 64 bits, 31 rounds and two different key sizes: 80 or 128 bits [20]. The characteristics of PRESENT was derived from the Serpent ciphers (non-linear substitution layer S-box) and DES (linear permutation layer p Layer). The S-box of PRESENT is shown in Table 1.

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S(x)$ | C | 5 | 6 | B | 9 | 0 | A | D | 3 | E | F | 8 | 4 | 7 | 1 | 2 |

4.2 GIFT
GIFT lightweight cipher is an improved version of PRESENT with an increased efficiency [23]. It overcomes the well-known weakness of PRESENT. GIFT is very simple and clean design that performs well for round-based implementations and is one of the most energy efficient cipher. GIFT ensures excellent performances from area-optimized hardware implementations to very fast software implementation on high-end platforms. GIFT algorithm was proposed for two block sizes namely 64-bit and 128-bit. A key size of 128-bits is used in both the versions of the algorithm [23]. The S-box of GIFT is shown in Table 2.

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $G(x)$ | 1 | A | 4 | C | 6 | F | 3 | 9 | 2 | D | B | 7 | 5 | 0 | 8 | E |

4.3 LED
LED is based on a substitution-permutation network (SPN). It is a 64-bit block cipher and supports key lengths from 64 to 128 bits. 64-bit key LED (named LED-64) and 128-bit key LED (named LED-128) [24]. There are three different architectures for FPGA implementation namely round-based implementation, fully serialized implementation and novel architecture. The S-box of LED is shown in Table 3.
Table 3. S-box of LED

| x  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S(x) | C | 5 | 6 | B | 9 | 0 | A | D | 3 | E | F | 8 | 4 | 7 | 1 | 2 |

4.4 KLEIN

KLEIN is another lightweight block cipher designed for resource-constrained devices. KLEIN is based on SPN network. It offers more flexibility and it mainly focuses on software implementations. The S-box of KLEIN is shown in Table 4.

Table 4. S-box of KLEIN

| x  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S(x) | 7 | 4 | A | 3 | 9 | 1 | F | B | 0 | C | 5 | 2 | 6 | 8 | E | D |

4.5 RECTANGLE

RECTANGLE is a lightweight block cipher based on Substitution Permutation network. The substitution layer consists of 16 4x4 S-box and permutation layer involves three rotations. It achieves a very good security. The S-box of RECTANGLE is shown in Table 5.

Table 5. S-box of RECTANGLE

| x  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S(x) | 0 | 5 | C | A | 1 | E | 7 | 9 | B | 0 | 3 | D | 8 | F | 4 | 2 |

5. Proposed Work

S-box is the major component in every encryption algorithm. When it comes to resource-constrained devices, S-box consumes most of its area. This project concentrates on reducing the size of the S-box. To create a compact S-box architecture, Boolean S-box is first formed and then it is factorized. The combinational design of the proposed lightweight S-box offers hardware advantages namely compactness in terms of a smaller number of gates.

5.1 Compact PRESENT S-box

Taking the work from [20]-[21], S-box architecture can be made compact by reducing the number of gates used. Karnaugh map is the technique used to optimize the 4-bit PRESENT S-box. The advantage of writing expressions in Boolean form is that it is valid only for true values. It lends itself to manipulating propositions that are either true or false. Combination of karnaugh mapping and factorization is used to compact the S-box. Factorization reduces the number of gates. A reduced number of gates decreases considerably the cost of the hardware, reduces the heat generated by the chip and most importantly increases the speed.

S-box is solved for true and complement values and the optimum method is chosen. PRESENT S-box is simplified using k-map by solving for 1’s and it resulted in 26 AND gates and 17 OR gates as discussed in [20]. But this S-box can be further minimized and that design is proposed below.

Table 6. Boolean PRESENT S-box

| w' | \((ab) + (ac'd) + (ac'd') + (ab'cd')\) |
|----|-----------------------------------|
| x' | \((a'bc') + (acd) + (bcd') + (b'cd) + (ab'c'd')\) |
| y' | \((a'c') + (a'bd) + (ab'd') + (abc'd)\) |
| z' | \((ab'd) + (acd) + (a'cd') + (a'b'd') + (a'bc'd) + (ab'c'd')\) |
Table 7. Compact Boolean PRESENT S-box

| Output | Expression |
|--------|------------|
| $w'$   | $f_1 + (d f_5) + (a f_4) + (f_7 f_8)$ |
| $x'$   | $(b f_5) + f_6 + (b f_8) + (b' f_2) + (f_3 f_4)$ |
| $y'$   | $f_5 + (a' f_9) + (d' f_1) + (f_3 f_2)$ |
| $z'$   | $(d f_3) + f_6 + (a' f_8) + (d' f_7) + (f_5 f_9) + (f_1 f_4)$ |
| $w$    | $(w)'$    |
| $x$    | $(x)'$    |
| $y$    | $(y)'$    |
| $z$    | $(z)'$    |

Table 7 shows the K-map expressions after simplification. It can be observed that Table 6 requires 41 AND, 15 OR and 8 NOT gates whereas after performing factorization PRESENT S-box can be implemented using 24 AND, 15 OR and 8 NOT gates. The input bits are denoted as $\{a, b, c, d\}$ where $a$ is the most significant bit (MSB) and $d$ is the least significant bit (LSB) and the output bits are denoted as $\{w, x, y, z\}$ where $w$ is the MSB and $z$ is the LSB. Boolean S-box can be constructed using AND, OR and NOT gates.

In table 6 there are some common factors among the expressions. This in turn reveals that power is wasted unnecessarily in computing the same values again and again and it also involves same hardware setup in multiple spaces. In order to avoid such power consumption, area, memory and delay the same factors can be represented as functions and can be used in the appropriate places. So, all such common factors among the expressions are represented as functional blocks. Each and every functional block is calculated only once. This leads to reduced computation time. This also leads to compact hardware structure.

5.2 Compact LED S-box

LED uses the same S-box as in PRESENT. The reason for using the same S-box is that the PRESENT S-box has very good cryptographic properties and small hardware footprints. Since both LED and PRESENT shares the same S-box there are some differences in the remaining process. The main difference between LED and PRESENT is the permutation layer. The permutation layer of LED is close as that of AES permutation layer whereas the permutation layer of PRESENT is entirely different and it is bit oriented [21]. Since the S-box remains the same, the proposed PRESENT S-box in section 5.1 can be used for LED.

5.3 Compact KLEIN S-box

Table 8. Boolean KLEIN S-box

| Output | Expression |
|--------|------------|
| $w$    | $(a' b' c') + (b c' d') + (b c d') + (a c' d')$ |
| $x$    | $(a' b' c') + (b' c' d') + (b c' d') + (a b c) + (a c d)$ |
| $y$    | $(a b' d') + (a' c d') + (b c' d') + (a b' d) + (a b' c)$ |
| $z$    | $(a c' d') + (a' b' c') + (a' b d') + (a b c) + (a b c' d) + (a b' c d)$ |
Table 9. Compact Boolean KLEIN S-box

| Output | Expression |
|--------|------------|
| w      | $f^9 + f^1 + (bf^6) + (af^4)$ |
| x      | $(c^1f^2) + (b^1f^4) + f^1 + f^8 + (cf^3)$ |
| y      | $(d^1f^2) + (a^1f^6) + f^1 + f^10 + (b^1f^5)$ |
| z      | $(a^1f^4) + (c^1f^7) + (d^1f^7) + f^8 + (c^1f^10) + (df^9)$ |

It can be observed that the k-map expression for KLEIN in Table 8 requires 42 AND gates, 16 OR gates and 4 NOT gates whereas the expressions in Table 9 requires 24 AND gates, 16 OR gates and 4 NOT gates. We received reduced number of gates.

5.4 Compact GIFT S-box

Table 10. Boolean GIFT S-box

| Output | Expression |
|--------|------------|
| w      | $(a'd) + (b'c'd) + (abc) + (acd')$ |
| x      | $(bc'd') + (a'bc') + (ab'd) + (acd) + (a'b'c)$ |
| y      | $(a'c'd) + (a'bd) + (ab'd') + (acd)$ |
| z      | $(a'bd) + (a'bc) + (ab'd) + (abc) + (abc'd') + (a'b'c'd')$ |

The optimum output expressions is chosen using k-map. Factorization is then performed to make a compact Boolean GIFT S-box. Factorization replaces the common factor by a function and uses it in the appropriate places. The compact GIFT S-box is shown below,

Table 11. Compact Boolean GIFT S-box

| Output | Expression |
|--------|------------|
| w      | $f^1 + (b'c'd) + (bf^2) + (df^2)$ |
| x      | $f^9 + (c^1f^4) + f^6 + f^3 + (cf^7)$ |
| y      | $(c^1f^1) + (d^1f^4) + (df^5) + f^3$ |
| z      | $(df^4) + (cf^4) + f^6 + (cf^5) + (af^9) + (f^7f^8)$ |

In the above expressions there are nine common factors and so nine functions are created. As a result, Boolean S-box can be constructed using reduced number of AND gates and OR gates. It can be observed that the k-map expression in Table 10 requires 39 AND gates, 15 OR gates and 4 NOT gates whereas the expressions in Table 11 requires 23 AND gates, 15 OR gates and 4 NOT gates.
5.5 Compact Rectangle S-box

**Table 12.** Boolean RECTANGLE S-box

|    | Expression                                      |
|----|------------------------------------------------|
| w  | \( (b'cd) + (a'b'c) + (a'bd) + (ac'd') + (bc'd) \) |
| x  | \( (a'b'd') + (a'b'c') + (bc'd) + (bcd') + (a'bc'd) \) |
| y  | \( (b'c'd') + (a'b'd') + (bc'd) + (abd) + (a'b'cd) + (a'bc'd) \) |
| z  | \( (ab'd') + (ab'c) + (a'bd') + (ab'c) + (abc'd) + (a'bc'd) \) |

The compact S-box is shown below,

**Table 13.** Compact Boolean RECTANGLE S-box

| Output | Expression                                      |
|--------|------------------------------------------------|
| w      | \( f3 + (cf4) + (a'f1) + (af9) + f2 \)          |
| x      | \( (d'f4) + f5 + f2 + (bf10) + (af3) \)         |
| y      | \( (b'f9) + f7 + f2 + (af1) + (a'f3) + (f8f10) \) |
| z      | \( f7 + (cf6) + (d'f8) + (cf8) + (af2) + (df5) \) |

It can be observed that the k-map expression in Table 12 requires 49 AND gates, 18 OR gates and 4 NOT gates whereas the expressions in Table 13 requires 25 AND gates, 18 OR gates and 4 NOT gates.

### 6. Results and discussion

Boolean S-box is created using K-map and further factorization is done to remove redundant expression. The result is a set of expression that consumes minimal number of AND gates and OR gates when compared to the non-factorized expressions gate counts. When comparing to the earlier works where only PRESENT was considered, the gate count and the delay is almost equal. Calculated delay from the earlier proposed work is 5.705 ns. Here in addition we made an effort to reduce the size of the S Box of LED, KLEIN, RECTANGLE and GIFT.

XILINX ISE is used to simulate the verilog code created for PRESENT S-box. The simulation result of PRESENT, KLEIN S-box is shown in Fig. 2 and 3.

**Figure 2.** Simulation of compact PRESENT S-box
Figure 3. Simulation of compact KLEIN S-box

Likewise the lightweight ciphers LED, RECTANGLE and GIFT are tested and verified. We made a comparison of all SOP and POS terms for the ciphers under consideration.

Table 14. Comparison of number of gates in SOP and POS for various cipher

| Algorithm | True output(SOP) | Complemented output(POS) |
|-----------|-----------------|--------------------------|
|           | Before factorization | After factorization | Before factorization | After factorization |
|           | AND | OR | NOT | AND | OR | NOT | AND | OR | NOT |
| PRESENT   | 45  | 17 | 4   | 26  | 17 | 4   | 41  | 15 | 8  |
|           |     |    |     | 24  | 15 | 8   |     |    |    |
| LED       | 45  | 17 | 4   | 26  | 17 | 4   | 41  | 15 | 8  |
|           |     |    |     | 24  | 15 | 8   |     |    |    |
| GIFT      | 39  | 15 | 4   | 23  | 15 | 4   | 41  | 15 | 8  |
|           |     |    |     | 26  | 15 | 8   |     |    |    |
| KLEIN     | 42  | 16 | 4   | 24  | 16 | 4   | 42  | 16 | 8  |
|           |     |    |     | 21  | 16 | 8   |     |    |    |
| RECTANGLE | 49  | 18 | 4   | 25  | 18 | 4   | 48  | 18 | 8  |
|           |     |    |     | 25  | 18 | 8   |     |    |    |
Table 15. Selection of optimum method for compact S-box

| Algorithm | Original S-box | Proposed compact S-box |
|-----------|----------------|------------------------|
|           | AND  | OR  | NOT | Total no. of gates | AND  | OR  | NOT | Total no. of gates |
| PRESENT   | 41   | 15  | 8   | 64              | 24   | 15  | 8   | 47              |
| KLEIN     | 42   | 16  | 4   | 62              | 24   | 16  | 4   | 44              |
| LED       | 41   | 15  | 8   | 64              | 24   | 15  | 8   | 47              |
| GIFT      | 39   | 15  | 4   | 58              | 23   | 15  | 4   | 42              |
| RECTANGLE | 49   | 18  | 4   | 71              | 25   | 18  | 4   | 47              |

Table 16. Comparison of parameters among the proposed compact S-box

| Algorithm | Total no. of gates | Delay (ns) | LUTs | Bonded IOBs |
|-----------|--------------------|------------|------|-------------|
| PRESENT   | 47                 | 5.755      | 17   | 21          |
| KLEIN     | 44                 | 5.705      | 14   | 18          |
| LED       | 47                 | 5.755      | 17   | 21          |
| GIFT      | 42                 | 5.705      | 13   | 17          |
| RECTANGLE | 47                 | 5.714      | 14   | 18          |

7. Conclusion

We have presented the compact implementation of PRESENT, GIFT and LED, KLEIN & RECTANGLE S-box. Karnaugh mapping is used to implement Boolean S-box. Further factorization is used to avoid redundant expressions. Redundant expressions consume the unwanted area, power and memory. Here we have represented the redundant expressions as functions wherever needed. These functions allow the redundant expression to be computed only once. On an average 20% reduction in space when compared to original S Box implemented using AND, OR & NOT gates can be achieved by this method. The proposed method is not a generic solution. It is to be designed for a particular cipher hence as our future work we are concentrating on a generic solution with security features against physical attacks.
References

[1] Ray S, Jin Y, and Raychowdhury A, The Changing Computing Paradigm with Internet of Things: A Tutorial Introduction 2016, IEEE Des. Test, vol. 33, no. 2, pp. 76–96, doi: 10.1109/MDAT.2016.2526612.

[2] Bohan Z, Xu W, Kaili Z, and Xueyuan Z, Encryption node design in internet of things based on fingerprint features and CC2530, 2013, Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013, pp. 1454–1457, doi: 10.1109/GreenCom-iThings-CPSCom.2013.256.

[3] Dhanda S S, Singh B, and Jindal P, Lightweight Cryptography: A Solution to Secure IoT, 2020, Wirel. Pers. Commun., vol. 112, no. 3, pp. 1947–1980, doi: 10.1007/s11277-020-07134-3.

[4] Prathiba A and Bhaaskaran V S K, Lightweight S-box architecture for secure internet of things, 2018, Inf., vol. 9, no. 1, doi: 10.3390/info9010013.

[5] Hwang D D et al., AES-based security coprocessor IC in 0.18-μm CMOS with resistance to differential power analysis side-channel attacks, 2006, IEEE Journal of Solid-State Circuits, vol. 41, no. 4, pp. 781–790, doi: 10.1109/JSSC.2006.870913.

[6] Bogdanov A et al., PRESENT: An Ultra-Lightweight Block Cipher, 2012, J. Empir. Theol., vol. 25, no. 2, pp. 127–152, doi: 10.1163/15709256-12341242.

[7] Mohamed K, Mohammed Pauzi M N, Hj Mohd Ali F H, Ariffin S, and Nik Zulkipli N H, Study of S-box properties in block cipher, 2014, in I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings, no. I4ct, pp. 362–366, doi: 10.1109/I4CT.2014.6914206.

[8] Kumar A and Tejani S, S-BOX Architecture, January, 2019, Springer Nat. Singapore, vol. 958, pp. 350–364, doi: 10.1007/978-981-3-3045-5.

[9] Çavuşoğlu Ü, Zengin A, Pehlivan I, and Kaçar S, A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system, 2017, Nonlinear Dyn., vol. 87, no. 2, pp. 1081–1094, doi: 10.1007/s11071-016-3099-0.

[10] Aboytes-González J A, Murguía J S, Mejía-Carlos M, González-Aguilar H, and Ramírez-Torres M T, Design of a strong S-box based on a matrix approach, 2018, Nonlinear Dyn., vol. 94, no. 3, pp. 2003–2012, doi: 10.1007/s11071-018-4471-z.

[11] Jamal S S, Attaullah, Shah T, AlKhaldi A H, and Tufail M N, Construction of new substitution boxes using linear fractional transformation and enhanced chaos, February 2019, Chinese J. Phys., vol. 60, pp. 564–572, doi: 10.1016/j.cjph.2019.05.038.

[12] Zhang X, Wu N, and Zheng X, The design method of compact composite field aes S-box based on AND-XOR array structure, 2018, in Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications, ICIEA 2017, vol. 2018-Febru, pp. 1881–1886, doi: 10.1109/ICIEA.2017.8283145.

[13] Ahmad N and Rezaul Hasan S M, Low-power compact composite field AES S-Box/Inv S-Box design in 65 nm CMOS using Novel XOR Gate, 2013, Integr. VLSI J., vol. 46, no. 4, pp. 333–344, doi: 10.1016/j.vlsi.2012.06.002.
[14] Morioka S and Satoh A, An Optimized S-Box Circuit Architecture for Low Power AES Design, 2003, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2523, pp. 172–186, doi: 10.1007/3-540-36400-5_14.

[15] Wong M M, Wong M L D, Nandi A K, and Hijazin I, Construction of optimum composite field architecture for compact high-throughput AES S-boxes, 2012, IEEE Trans. Very Large Scale Integr. Syst., vol. 20, no. 6, pp. 1151–1155, doi: 10.1109/TVLSI.2011.2141693.

[16] Prasetyo K N, Purwanto Y, and Darlis D, An implementation of data encryption for Internet of Things using blowfish algorithm on FPGA, 2014, 2014 2nd Int. Conf. Inf. Commun. Technol. ICoICT 2014, pp. 75–79, doi: 10.1109/ICoICT.2014.6914043.

[17] Yao X, Chen Z, and Tian Y, A lightweight attribute-based encryption scheme for the Internet of Things, 2015, Futur. Gener. Comput. Syst., vol. 49, pp. 104–112, doi: 10.1016/j.future.2014.10.010.

[18] Wood C A, Radziszowski S P, and Lukowiak M, Constructing large S-boxes with area minimized implementations, 2015-Decem, in Proceedings - IEEE Military Communications Conference MILCOM, vol., pp. 49–54, doi: 10.1109/MILCOM.2015.7357417.

[19] Prathiba A and Kanchana Bhaaskaran V S, Hardware footprints of S-box in lightweight symmetric block ciphers for IoT and CPS information security systems, 2019 May, Integration, vol. 69, no. pp. 266–278, doi: 10.1016/j.vlsi.2019.05.003.

[20] Tay J J, Wong M L D, Wong M M, Zhang C, and Hijazin I, Compact FPGA implementation of PRESENT with Boolean S-Box, 2015, in Proceedings of the 6th Asia Symposium on Quality Electronic Design, ASQED 2015, pp. 144–148, doi: 10.1109/ACQED.2015.7274024.

[21] Sbeiti M, Silbermann M, Poschmann A, and Paar C, Design space exploration of present implementations for FPGAS, 2009, in Proceedings - 2009 5th Southern Conference on Programmable Logic, SPL 2009, pp. 141–145, doi: 10.1109/SPL.2009.4914893.

[22] Yu X, Wu N, Zhou F, Zhang J, and Zhang X, A Compact Hardware Implementation for the SCA-resistant PRESENT Cipher, 2019- Octob, IECON Proc. (Industrial Electron. Conf., vol., pp. 5463–5468, doi: 10.1109/IECON.2019.8927097.

[23] Lara-Nino C A, Diaz-Perez A, Morales-Sandoval M, FPGA-Based Assessment of Midori and Gift Lightweight Block Ciphers, 2018, In: Naccache D. et al. (eds) Information and Communications Security. ICICS 2018. Lecture Notes in Computer Science, vol 11149. Springer, Cham. https://doi.org/10.1007/978-3-030-01950-1_45

[24] Kousalya R and Sathish Kumar G A, A Survey of Light-Weight Cryptographic Algorithm for Information Security and Hardware Efficiency in Resource Constrained Devices, 2019, Proc. - Int. Conf. Vis. Towar. Emerg. Trends Commun. Networking, ViTECoN 2019, pp. 1–5, doi: 10.1109/ViTECoN.2019.8899376.