J. Integr. Sci. Technol. 2022, 10(1), 39-43 Article

Journal of Integrated
SCIENCE & TECHNOLOGY

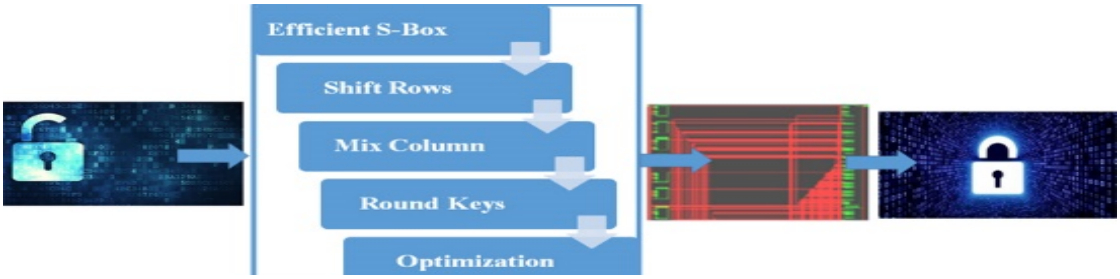
Design of efficient S-box for Advanced Encryption Standard

Sarita Sanap[1] Vijayshree More[2]

[1]Department of Electronics and Telecommunication Engineering,Maharashtra Institute of Technology, Dr.B.A.M.U, Aurangabad, India. [2]Department of Electronics and Computer Engineering, Jawaharlal Nehru Engineering College, MGM University, Aurangabad, India.

Received on: 10-Jan-2022, Accepted and Published on: 5-Mar-2022

ABSTRACT



In digital era, data security is a necessary requirement. To establish secure communication modern encryption techniques plays a vital role. By employing an efficient S-box constraints of area, power and speed are achievable. In this paper method for efficient S-box is presented which provides promising solution in terms of required constraints. Comparison of proposed method with other existing method is also done by implementing it on field programmable gate array .It shows that proposed method uses only 6.14% slices resulting 13% improvement in comparison with other methods. Reduction in LUTs are done by 12.42 % in proposed method. Thus optimization is achieved in terms of number of slices and number of LUTs. Delay and memory usage is also reduced significantly.

Keywords: S-box, AES, Galois field, residue prime numbers, Field programmable gate array

INTRODUCTION

Encryption techniques provides data integrity,non-repudiation and authentication over insecure channels. Digital technology adoption has increased, specially after outbreak of Covid-19.[1] Security has become key concern in this digital transformation. In most modern encryption techniques S-box is the first transformation. Substitution box maps n input bit to m output bits. It is important method to create confusion, which decides strength of ciphertext. As a result strength and reliability of encryption technique is dependent on S box.[2] In every round substitution is required so it is a time consuming process. S box is massive and requires more power than other components.

*Corresponding Author: Mrs. Sarita D Sanap, Assistant Professor, Department of Electronics and Telecommunication, Maharashtra Institute of Technology, Aurangabad.
Email: saritawagh1@gmail.com

Cite as: *J. Integr. Sci. Technol., 2022, 10(1), 39-43.*

 ISSN: 2321-4635 http://pubs.iscience.in/jist

In this work S-box in Advanced Encryption Standard (AES) is considered. In AES Implementation S box is crucial transformation. Using field programmable gate arrays the design constraints are achieved. In this paper efficient encryption method is proposed focusing on S-box using Galois field and residue prime numbers. Comparison with other method based on area, memory usage and delay parameter is also done.

Related work

Are ***Transformation used in AES***

AES is non-Feistel cipher defined in three versions with 10, 12, and 14 rounds, the key size is 128, 192, and 256 respectively.[3] Round key is always 128 bits. To obtain security AES uses four transformations byte substitution,shift rows,mix column and addroundkey as shown in figure 1. Each transformation takes a state and generates new state for next step and finally generates ciphertext.

Byte Substitution(S-box): In this process the state is considered as 4x4 matrix of bytes. Transformation is carried out on one byte at a time. Each byte is changed independently. Total 16 independent byte transformation takes place. In symmetric key algorithm S-box is most important component. In SubByte block transformation is

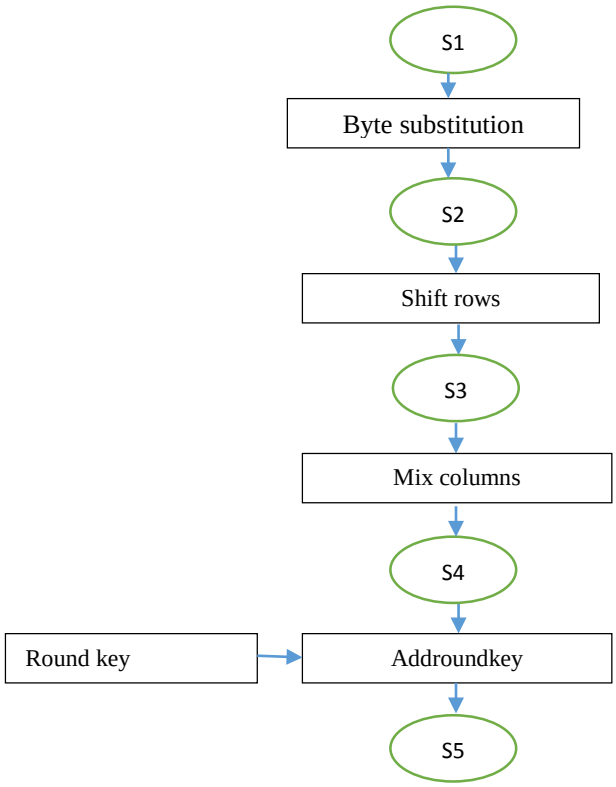


Figure 1.Transformations used in AES

done by using nonlinear transformation. On each byte SubByte transformation is performed. Using polynomial representations the SubByte Transformation can be implemented on hardware.[4] The software implementation can be done using 16x16 Look Up Table (LUT). SubByte transformation using Composite Field Arithmetic reduces the complexity and also adds pipelining concept in its structure. Realization of Composite field arithmetic can be done by using different irreducible polynomials. Total 16 parallel S-boxes each with 8 inputs and 8 outputs are required. The S-box operation is the only nonlinear transformation of AES algorithm.[5].An invertible map, fractional linear transformation on Galois field can be used for S-box generation.[6] LUTs in composite field arithmetic is carried out for S-box and multiplication process which decreases computation time and hardware complexity.[7] Combinational logic based pipelined S-box is implemented and masking is done by xor operation by the author.[8] Algorithm to generate orthmorphism over Galois field can be done to generate 8x8 S-box.[9] Instead of entire Galois field ,elements of multiplicative subgroup of Galois field helps in construction of S-box.[10] Internal parallelism is also achievable through residue number system. Residue number system is used in mixed radix system to represent integers.[11-12] Rotational symmetry of Galois field for inversion is used in implementation of S-box.[13] In cloud systems security is provided by residue number system by applying different keys.[14] Complete Latin square expansion gives a pair of orthogonal matrices which is used for S-box after Scrambling.[15] As S-box is non- linear component, it is mostly a primary target for side channel attacks.[16] So design of S-box is most crucial part of encryption technique. Coset diagrams and Fibonacci sequences can also be used for substitution.[17]

Shift rows: In this transformation rows of the state are shifted in cyclic manner over different offsets. It operates on one row at a time. In this operation row 0 is not shifted and last row shifted three times.[18]

Mix columns: New column is transformed from each column of the state. It is a matrix multiplication with modulus 10001101. The columns are multiplied modulo xl^4+1 with polynomial a1(xl) as polynomials over Galois Field (2^8), where xl is characteristics of the field. For mix columns the overall utilization of redundant bits is minimized in the boolean expression using resource sharing architecture and gate replacement technique, which helps to lower power consumption and cost of the encryption algorithm.[19]

Addroundkey: This transformation operates one column at a time. With each column matrix a round key word is added. XOR operation is performed on each column with key word.[20]

S-box Implementation using Galois field

In first stage multiplicative inverse is used to replace each byte. In Second stage invertible affine transformation is performed. Complexity is depend on representation of field elements. S box transformation is represented as a lookup table. Elements of Galois field are defined as equation 1.

$$Gf = (0, 1, 2, \ldots, xl-1) \cup (xl, xl+1, xl+2, \ldots, xl+xl-1) \cup (xl^2, xl^2+1, xl^2+2, \ldots, xl^2+xl-1) \cup \ldots \cup (xl^{n-1}, xl^{n-1}+1, xl^{n-1}+2, \ldots, xl^{n-1}+xl-1) \quad (1)$$

Multiplication of polynomial is given by eq. 2

$$M(xi) = (f(xi) \cdot g(xi)) \,(\mathrm{mod} m(xi)) \quad (2)$$

The multiplicative inverse of f(x) is given by a(x) such that it follows

$$(f(xi) \cdot a(xi)) \,(\mathrm{mod}\, m(xi)) = 1 \quad (3)$$

Where m(x) is an irreducible polynomial of degree at least n in GF. The modulo polynomial for finding the multiplicative inverse is

$$m(xi) = xi^{8} + xi^{4} + xi^{3} + xi^{1} + xi^{0} \quad (4)$$

Implementation results for S-box using Galois field

VLSI architecture implementation gives more security with cryptographic algorithm of S-box.[2] Hardware speed and software flexibility advantage is provided by FPGAs.As compared to ASIC FPGA gives advantage of reconfigurability and low design cost.[21] So S-box using Galois field is implemented using Xilinx ISE 14.7,Aldec HDL and Virtex 5 FPGA. The Virtex-5 consists of Block RAM/first in first out (FIFO), DSP slices, Select IO technology, DCM and PLL clock generators, and other advanced configuration option. This implementation uses 521 slices and 953 input LUTs.

Implementation of S Box using Galois Field ROM datatype

In this method S box contains pre computed value stored at predefined address. This alternative memory architecture is very much optimized. As the entire table entries are fixed in the lookup table, use of ROM is preferred.[22] The architecture needs small ROM modules in place of one large module, since each lookup entry is based on a maximum of 8-bit address, which translates to 256 entries which are based on Galois field. Primitive polynomial of GF(2^4) is used as reduction of GF(2^8).[23] Implementation is done on same configuration but as ROM datatype is used this design occupies 64 slices and 128 input LUTs.

PROPOSED METHODOLOGY

Implementation of S-BOX for AES using reduced prime numbers is proposed in this work. For comparative analysis S- box using GF (2^8) is also implemented by two methods. Implementation of these algorithms and comparative study is done. Design implementation, simulation and synthesis is done on Field programmable Gate array virtex 5 using Xilinx 14.7 ISE and Aldec active HDL.

Structure of each round and subbyte transformation

Each round of AES except the last uses four invertible transformations. This transformation is implemented for AES-128 as shown in figure 2. S-Box is a substitution function whose nonlinearity help to protect against linear cryptanalysis. In this work focus is on efficient design of S-box so that constraints are mapped. Substitution implementation is done by Shannon's confusion-diffusion principles. Confusion is most desirable requirement of encryption algorithm. A strong confusion property is required to secure data in the form of ciphertext. Diffusion is randomness required in substitution, for change in one bit of input, the substituted output should be changed by at least half of bits. This makes it unpredictable and hence more secure communication is possible.

Algorithm 1 : Pseudocode for AES

```
State=n
Si=Addroundkey (state, w (4))
for i in range 1 to 9
Sub Bytes (si)
Shift Rows (si)
Mix Columns (si)
AddRoundKey (si, w[4*i,4*i+3])
end for
end
```

Each transformation forms a state which is used for next round. Last round uses only three transformation. Pseudocode for this design is presented in algorithm 1. For loop is excuted 10 times for 128 bits.This transformation operated separately on every byte of the state which uses 16-byte (128-bit) S-Boxes.

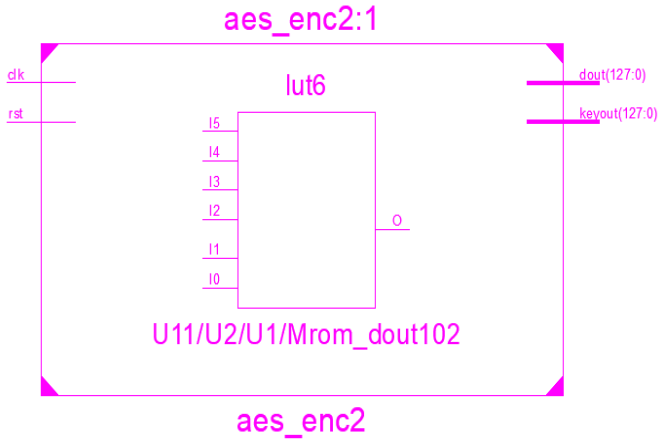


Figure 2.S-box implementation using Aldec HDL

Implementation of S-box using proposed residue prime number system

Residue number system is given by set of N co prime modulli (m_1, m2,m3….m_n).A number x is given in N channel as given in equation 5.

$$X=\{x_1, x_2, x_3,\ldots x_n\} \quad (5)$$

Where xi is X's residual in relation to m_i. Due to this it has digit sets as

$$Sets=\{[0,ni_1\text{-}1],[0,ni_2\text{-}1],[0,ni_3\text{-}1]\ldots.[0,ni_n\text{-}1]\} \quad (6)$$

Residue prime arithmetic shows that It gives a multiplicative inverse in the modulo P field. As a result, instead of using the Galois field to compute S-Box entries, the field of residues modulo prime can be used. A residue number system have pairwise coprime moduli set. A lookup table having 256 entries is formed to substitute Galois field lookup table.

Algorithm 2: Subbyte routine

```
sub Bytes(S)
for k in range (len (S)):
S [k] = sbox [S[k]]
a=bytemulinv
bytematrix (a, b)
for k in range(len(S))
c=bk⊕b(k+4)mod8⊕b(k+5)mod8⊕b(k+6)mod8⊕b(k+7)mod8
d=ck ( bytematix)
end for
matrixbyte (d,d)
byte =d
end
```

Substitution process is presented by algorithm 2. S represents state in the algorithm. It is implemented on FPGA using ROM component.

Implementation of S-box using proposed optimized residue prime number system

Reduction of 256 entries are done in this proposed method. Reduction is based on equation 7 and equation 8 where i, j, m and n presents residue prime numbers. As the numbers are inverse of each other, only number or its inverse is considered for the record of S-box. This entries are unidentified, so more confusion to the S-Box implementation is created which is not present in Galois Field based S-Box.

$$S(i, j) = m, n \quad (7)$$

$$S(m, n) = i, j \quad (8)$$

To make it more secure a seed value function is applied on each entry.This process is presented using algorithm 3.

Algorithm 3: subbyte process based on optimized RNS

```
Step 1: Initialize Fs_boxin
Step 2: activate event on clock
Step 3: define ram type array (natural range< >)
Step 4: store reduced sbox_ram: ram_type (127 down to 0)
Step 5:sbox_ram⊕seed value
Step 6: byteoutput <= sbox_ram
Step 7: end
```

By following process shown in algorithm 3 final LUT is formed as shown in Table 1. In Galois field based S-box LUT consists of 256 values where as proposed LUT needs only 129 values. S-box using this look up table is implemented on FPGA.Simulation waveforms obtained are shown in figure 3. It gives significant reduction in number of slices and bonded IOBs. This results in optimization of resources.More randomness is obtained by ex-oring of seed value .

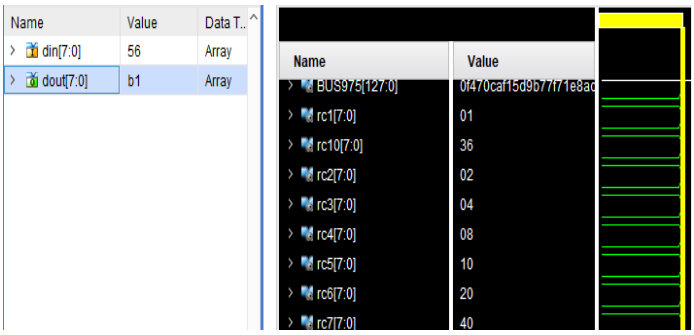


Figure 3. Simulation output waveforms of proposed system

Table 1. Optimized residue prime number based LUT

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	af	ae	2e	59	6e	c8	84	3c	4e	67	1b	14	39	1d	65	d7
1	5e	d6	cb	49	f5	9e	71	11	e4	e7	f6	41	ca	c6	93	68
2	56	3b	12	44	9d	2b	dc	3e	82	c0	36		c0		c0	00
3	90			d1	20	ce	d8	5c	1c	57	4d	92			4b	c9
4	52	f8	e5	45	70	3a	59	1a		60			15	58	66	5b
5	38	a0	d2	7d	62	d0			17			7e	1f	37	77	
6				28					78	b0	1e	5a	13	4f	55	
7	75	db	d3			29	30				31	23		73		
8	50		30	61		20			5f	5d	64	cd		3f	74	
9			7a								53	6d	4a	40		
a	63	10	7c					42	48		6f	51				
b							46	10		47						
c					69	4c										72
d	43				76		54									
e																
f																

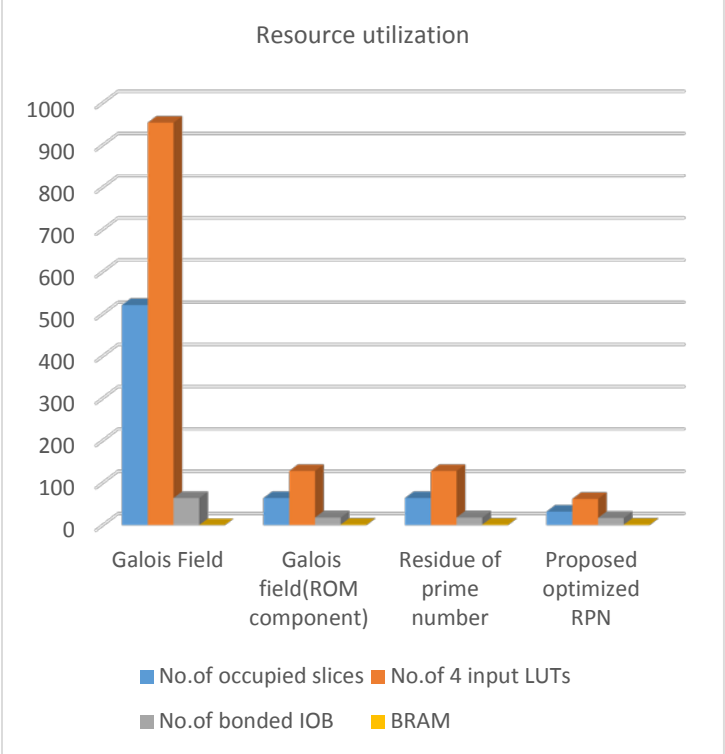
RESULTS AND DISCUSSIONS

Simulation and Implementation results for all methods and proposed optimized method are analyzed. Resource utilization is as shown table 2. It elaborates the overall device utilization including: number of slices, number of LUTs, and number of bonded IOBs and block RAMs.

Table 2. Resource utilization

Parameters	Galois Field	Galois field(ROM component)	Residue of prime number	Proposed optimized RPN
No.of occupied slices	521	64	64	32
No.of 4 input LUTs	953	128	128	62
No.of bonded IOB	64	18	18	17
BRAM	0	1	1	1

In comparison with S box using Galois field, Galois field (ROM component) and Residue of prime number occupies 12% slices which implies that less area is used in these implementation. Significantly it shows that proposed optimized method uses only 6.14% slices. This results in less area requirement and consequently less power requirement. Proposed method gives reduction of LUTs. Only 6% LUTs are needed in optimized RPN as compared to other methods. 26% less numbers of bonded IOBs are used in proposed work as compared to Galois field implementation.



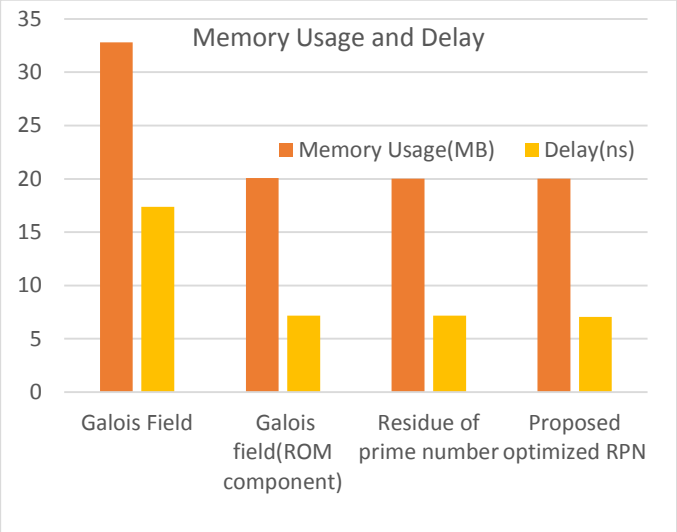
Graph 1.Resource utilization

As per graph 1 for resourse utilization,S-box based on optimzed residue prime number utilizes least number of occupied slices.Number of LUTs are also reduced significantly. Memory usage and delay parameters are also analyzed using xilinx 14.7. It is given in table 3.

Table 3.Memory Usage and delay

Parameters	Galois Field	Galois field(ROM component)	Residue of prime number	Proposed optimized RPN
Memory Usage(MB)	32.79	20.08	20.017	20.009
Delay(ns)	17.386	7.165	7.16	7.06

Comparison of memory usage and delay obtained is shown in graph 2. Synthesis report gives total logic and route delay.



Graph 2.Memory usage and delay

Due to reduced number of LUT and occupied substantial reduction in memory is observed in S-box based on optimized RPN. Delay observed is also reduced in significant way which helps high speed execution of Advanced Encryption Standard.

CONCLUSIONS

In encryption algorithms S-box is an important component. In this paper optimized residue of prime number S-box for AES is proposed. Proposed method gives 13% improvement in terms of occupied slices compared to other methods. Required LUTs are optimized by 12.42%. Comparative analysis with other method elaborates that design and implementation using proposed method results in less utilization of resources and memory. So constraint of area, delay and memory usage is achieved efficiently.

CONFLICT OF INTEREST

Authors declared no conflict of interest.

REFERENCES

1. B. Sowmiya, V.S. Abhijith, S. Sudersan, et al. A Survey on Security and Privacy Issues in Contact Tracing Application of Covid-19. *SN Comput. Sci.* **2021**, 2 (3), 1–11.
2. A. Reyhani-Masoleh, M. Taha, D. Ashmawy. New Low-Area Designs for the AES Forward, Inverse and Combined S-Boxes. *IEEE Trans. Comput.* **2020**, 69 (12), 1757–1773.
3. J. Daemen, V. Rijmen. The Design of Rijndael. *New York* **2002**, 255. Springer.
4. D.Q. Huy, N.M. Duc, L.D. Khai, V.D. Lung. Hardware Implementation of AES with S-Box using Composite-field for WLAN Systems. *2019 IEEE-RIVF Int. Conf. Comput. Commun. Technol.* **2019**, 1–6.
5. Data Communications and Networking By Behrouz A.Forouzan.
6. S. Farwa, T. Shah, L. Idrees. A highly nonlinear S-box based on a fractional linear transformation. *Springerplus* **2016**, 5 (1), 1658.
7. C. Arul Murugan, P. Karthigaikumar, S. Sathya Priya. FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications. *Automatika* **2020**, 61 (4), 682–693.
8. L. S-box. Hardware Implementation of AES Algorithm with. **2017**, 26 (9), 1–19.
9. H. Han, S. Zhu, Y. He, et al. Evolutionary generation of orthomorphisms in the Galois field F28. *Evol. Intell.* **2020**, No. 0123456789.
10. T. Shah, A. Qureshi. S-box on subgroup of galois field. *Cryptography* **2019**, 3 (2), 1–9.
11. K. Bigou, A. Tisserand. Hybrid Position-Residues Number System. *Proc. - Symp. Comput. Arith.* **2016**, 2016-Septe (2), 126–133.
12. S. Kawamura, Y. Komano, H. Shimizu, T. Yonemura. RNS Montgomery reduction algorithms using quadratic residuosity. *J. Cryptogr. Eng.* **2019**, 9 (4), 313–331.
13. F. Wegener, L. De Meyer, A. Moradi. Spin Me Right Round Rotational Symmetry for FPGA-Specific AES: Extended Version; Springer US, **2020**; Vol. 33.
14. D. Schoinianakis. Residue arithmetic systems in cryptography: a survey on modern security applications. *J. Cryptogr. Eng.* **2020**, 10 (3), 249–267.
15. Z. Hua, J. Li, Y. Chen, S. Yi. Design and application of an S-box using complete Latin square. *Nonlinear Dyn.* **2021**, 104 (1), 807–825.
16. H. Li, G. Yang, J. Ming, Y. Zhou, C. Jin. Transparency order versus confusion coefficient: a case study of NIST lightweight cryptography S-Boxes. *Cybersecurity* **2021**, 4 (1), 1–20.
17. I. Shahzad, Q. Mushtaq, A. Razaq. Construction of New S-Box Using Action of Quotient of the Modular Group for Multimedia Security. *Secur. Commun. Networks* **2019**, 2019.
18. S. Chen, W. Hu, Z. Li. High Performance Data Encryption with AES Implementation on FPGA. *Proc. - 5th IEEE Int. Conf. Big Data Secur. Cloud, BigDataSecurity 2019, 5th IEEE Int. Conf. High Perform. Smart Comput. HPSC 2019 4th IEEE Int. Conf. Intell. Data Secur.* **2019**, 149–153.
19. S. Madhavapandian, P. MaruthuPandi. FPGA implementation of highly scalable AES algorithm using modified mix column with gate replacement technique for security application in TCP/IP. *Microprocess. Microsyst.* **2020**, 73, 102972.
20. P. Wang, Y. Zhang, J. Yang. Research and design of AES security processor model based on FPGA. *Procedia Comput. Sci.* **2018**, 131, 249–254.
21. K. Shahbazi, S.B. Ko. High throughput and area-efficient FPGA implementation of AES for high-traffic applications. *IET Comput. Digit. Tech.* **2020**, 14 (6), 344–352.
22. R.A. Padmavathi, K.S. Dhanalakshmi. An Advanced Encryption Standard in Memory (AESIM) Efficient, High Performance S-box Based AES Encryption and Decryption Architecture on VLSI. *Wirel. Pers. Commun.* **2021**, No. 0123456789.
23. M. Qasaimeh, R.S. Al-Qassas, M. Ababneh. Software design and experimental evaluation of a reduced aes for iot applications. *Futur. Internet* **2021**, 13 (11), 1–21.