Security Analysis against Differential Cryptanalysis using Active S-Boxes

R.Kousalya[1], Dr.G.A.Sathish Kumar[2]

[1,2]Department of Electronics and Communication Engineering
Sri Venkateswara College of Engineering, Sriperumbudur Tk – 602 117, Kancheepuram District, Tamil Nadu, India

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract

Cryptography is a technique that uses mathematics for encryption of original data followed by decryption of encrypted data. Cryptography enables the impregnability of information while it passes through the Internet or wireless medium with more concern for security. It is highly challenging to execute the conventional cipher technique algorithms in a resource-constrained environment attributable to their size, speed and throughput. The Lightweight cipher technique is an algorithm that features a low footprint and computational complexity. In cipher methods, the substitution box acts major role in the encryption/decryption algorithm. The procedure of creating powerful S-boxes in cryptographic algorithm never end. Various methods are opened for creating a powerful S-box leading difficulty for an attack. This paper depicts a key-based dynamic Ssubstitution box for every round of encryption in PRESENT algorithm. Differential Cryptanalysis attack is verified with the quantity of active S-boxes in each round.

Keywords: — Key dependent S-box, Key Scheduling, Light-weight cryptography, PRESENT algorithm, Security, S-box.

1. Introduction

Light-weight cipher technique [1] is an encryption/decryption method that technique a tiny footprint and/or low computational difficulty. It is targeted for constrained devices that include RFID tag, sensors, so on. The inspiration of this cipher technique is to use minimal memory and fewer power supplies to deal the security that can work over resource-constrained devices. The lightweight cipher technique is expected with compact and high speed than traditional cryptography. Its drawback is less secured. The optimizing encryption algorithm supports the standard cryptographic primitives to run on portable and resource-constrained devices [2].

The component that are necessary for hardware execution are cited below

- Size
- Power consumption
- Throughput and Delay

The foremost factor is highly required as these algorithms meant for resource limited devices. Power is a crucial metrics considered for energy harvesting devices while the power utilization is important with battery-operated devices. A high throughput is critical for hardware with enormous data transmissions namely camera or a vibration sensor, while high speed is vital for the real-time control processing of car-control system, etc.

The important part in Light-weight algorithm is an S-box. S-box is the core and nonlinear component of cipher technique. Consequently, it is said that the cipher technique's strength is decided by S-box. A good S-box will have the following properties [3].

- Robustness
- Balancing
- Avalanche effect
- Nonlinearity
- Differential Uniformity

Substitution is a nonlinear alteration in which bits are shuffled. The property creates the complex uniqueness between the encrypted output and key is called confusion. This property makes it tough to trace the key out of encrypted data and if a small change happens in single key bit, it becomes challenging for retrieving ciphertext.

Section 2 explains the PRESENT algorithm, section 3 deals the Key-dependent S-box algorithm section 4 illustrates the features of differential cryptanalysis based on active s-boxes of the cited algorithms, while section 5 shows the simulated results of Lightweight Cryptographic algorithm and section 6 deals the conclusions related to this work.

2. PRESENT Algorithm

The PRESENT algorithm [4] consists of Substitution and Permutation block. It is termed as SPN based algorithm takes a same secrete key at the transmission and reception side. This algorithm allowed performing in devices with minimal-sized hardware which gives a sense of adequate use of resources making it easy to utilize the processors with small bus. Figure 1 shows the block diagram of PRESENT algorithm.

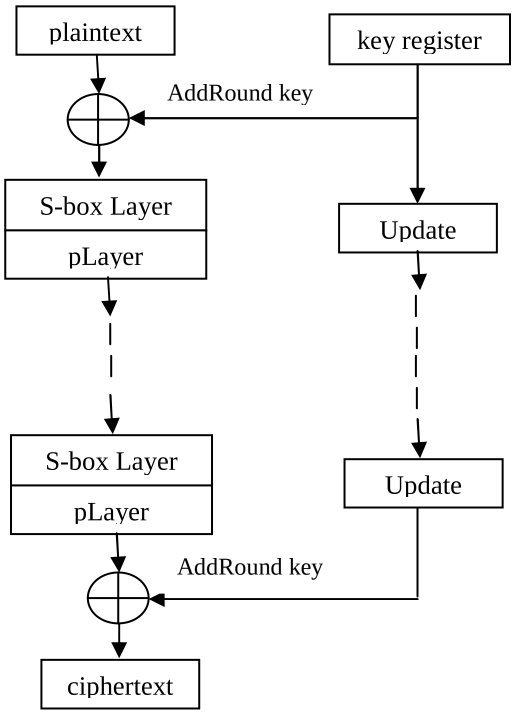

Figure 1 PRESENT Algorithm

2.1 AddRound Key

In the AddRound Key, Exclusive-OR operation is performed with key and state. In each round, 64-bits key is produced using scheduling of main key as same size as state. The 64-bits key is Exclusive ORed with state. XOR operation is applied between 64-bit plaintext and 64-bit sub key that is extracted from 80-bit key. Sub key is extracted from 64 leftmost bits of current content of register K as mentioned below.

$$K_i = k_{63}\, k_{62} \ldots k_0 = k_{79}\, k_{78} \ldots k_{16}\,, \quad 1 < i < 32$$

Round key $K_i = k_{i63} \ldots k_{i0}$ for i ranges from 1 to 32 and current state $b_{63} \ldots b_0$, AddRound Key takes the operation for j ranges from 0 to 63,

$$b_j = b_j \oplus k_{ij}$$

2.2 Substitution Box (S-Box)

The S-box used in PRESENT algorithm is 4 x 4 S-box.The action of S-box in hexadecimal notation is given in Table 1.

Table 1 S-box of PRESENT algorithm

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

In each round, S-Box Layer takes the state $b_{63} \ldots b_0$ as 16 Four-bit words $w_{15} \ldots w_0$ where $w_i = b_{4*i+3}||b_{4*i+2}||b_{4*i+1}||b_{4*i}$ for i ranges from 0 to 15. The output nibble $S[w_i]$ provides the updated state values in the obvious way.

2.3 Permutation (PLayer)

The 64- bit permutation utilized in PRESENT algorithm. Bit on position i of current state is moved to bit on position P(i). It is noted that the bit on position 0 of input moved to the PLayer position 0 and bit on position 1 moved to bit position 4 of PLayer, so on.

$$P(i) = i.16 \bmod 63,\ 0 < i < 63$$

Table 2 Permutation of PRESENT algorithm

i	P(i)	i	P(i)	i	P(i)	i	P(i)
0	0	16	4	32	8	48	12
1	16	17	20	33	24	49	28
2	32	18	36	34	40	50	44
3	48	19	52	35	56	51	60
4	1	20	5	36	9	52	13
5	17	21	21	37	25	53	29
6	33	22	37	38	41	54	45
7	49	23	53	39	57	55	61
8	2	24	6	40	10	56	14
9	18	25	22	41	26	57	30
10	34	26	38	42	42	58	46
11	50	27	54	43	58	59	62
12	3	28	7	44	11	60	15
13	19	29	23	45	27	61	31
14	35	30	39	46	43	62	47
15	51	31	55	47	59	63	63

Table 2 express the permutation of PRESENT algorithm. 64-bits generated from substitution box are rearranged bit-by-bit position. A good PLayer should support the diffusion property. The diffusion property states that making the complex relationship between plaintext and ciphertext.

2.4 Key Schedule

PRESENT algorithm takes the key size either 80 bits or 128 bits. Here 80-bit of key is considered. The steps to be followed for key updation in the Key register, $K = k_{79}\, k_{78} \ldots k_1\, k_0$ is shown below

- $[k_{79}k_{78} \ldots k_1\, k_0] = [k_{18}k_{17} \ldots k_{20}\, k_{19}]$ -> 61 bit circular Left shift
- $[k_{79}\, k_{78}\, k_{77}\, k_{76}] = S\, [k_{79}\, k_{78}\, k_{77}\, k_{76}]$
- $[k_{19}k_{18}k_{17}k_{16}k_{15}]=[k_{19}k_{18}k_{17}k_{16}k_{15}]$^roundcounter

The key register is circularly left shifted by 61 bit positions, the left-most 4-bits are fed into the substitution box of PRESENT algorithm and roundcounter value i is XORed with bits $k_{19},k_{18},k_{17},k_{16},k_{15}$ of K with the LSB bit of roundcounter. The scheduling process is repeated for each round. Figure. 2 shows Key scheduling output of 80-bit key.

3. PRESENT Algorithm with Key Dependent S-Boxes

The key dependent S-box [5] is in line to PRESENT algorithm where it has 32 rounds, 64-bit plaintext with 80-bit of key. The main difference is that 16 S-boxes are used in key dependent S-box wherein in PRESENT algorithm only one S-box is used.

Figure 3 depicts the PRESENT algorithm with key dependent S-box. Selection function is acquired by taking XOR operation of key elements and selects single S-box out 16 S-boxes.

```
Key scheduling:
Roundkey for round  0  :  1111222222333334
Roundkey for round  1  :  3888822224444446
Roundkey for round  2  :  7cccc71110444489
Roundkey for round  3  :  b1110f9998e22209
Roundkey for round  4  :  6112362221f3331e
Roundkey for round  5  :  48882c2246c4443c
Roundkey for round  6  :  eccc6911058448db
Roundkey for round  7  :  d110fd998d2220b3
Roundkey for round  8  :  a1237a221fb331a0
Roundkey for round  9  :  c882d4246f4443f2
Roundkey for round  10  :  0cc699105a848ded
Roundkey for round  11  :  810fc198d3220b55
Roundkey for round  12  :  f237b021f8331a62
Roundkey for round  13  :  982d5e46f6043f00
Roundkey for round  14  :  5c699305abc8dec7
Roundkey for round  15  :  20fc0b8d3260b57e
Roundkey for round  16  :  737b041f8171a644
Roundkey for round  17  :  32d5ee6f6083f026
Roundkey for round  18  :  7699065abdcdec19
Roundkey for round  19  :  6fc08ed320cb57b0
Roundkey for round  20  :  c7b06df811da6413
Roundkey for round  21  :  ad5ed8f60dbf0231
Roundkey for round  22  :  b99055abdb1ec1bc
Roundkey for round  23  :  ec08d7320ab57b68
Roundkey for round  24  :  cb06fd811ae6415a
Roundkey for round  25  :  05edb960dfb02350
Roundkey for round  26  :  c90560bdb72c1bfb
Roundkey for round  27  :  a08d5920ac17b6e8
Roundkey for round  28  :  c06ff411ab24158c
Roundkey for round  29  :  eedbb80dfe82356a
Roundkey for round  30  :  90563ddb7701bfdf
Roundkey for round  31  :  18d5b20ac7bb6eef
```

Figure 2 Key Scheduling of 80-bit key

The S-box chosen in a round is decided by key calculated from 80-bit input key. It is proposed to pick the substitution box using 4 bit key in a round. S_0 to S_{15} S-boxes are taken from Serpent and Humming bird algorithm [6].

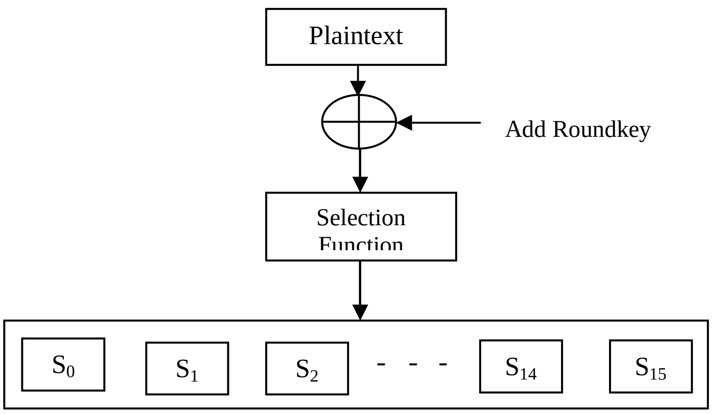

Figure 3 Key Dependent S-Box

Table 3 indicates the S-boxes used in PRESENT algorithm with key dependent technique. If 4-bit key is 0101, then the algorithm selects S_5 box in that round. Selection of S-box varies across each round.

Table 3 S-boxes in Key dependent S-Box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	3	8	F	1	A	6	5	B	E	D	4	2	7	0	9	C
$S_1(x)$	F	C	2	7	9	0	5	A	1	B	E	8	6	D	3	4
$S_2(x)$	8	6	7	9	3	C	A	F	D	1	E	4	0	B	5	2
$S_3(x)$	0	F	B	8	C	9	6	3	D	1	2	4	A	7	5	E
$S_4(x)$	1	F	8	3	C	0	B	6	2	5	4	A	9	E	7	D
$S_5(x)$	F	5	2	B	4	A	9	C	0	3	E	8	D	6	7	1
$S_6(x)$	7	2	C	5	8	4	6	B	E	9	1	F	D	3	A	0
$S_7(x)$	1	D	F	0	E	8	2	B	7	4	C	A	9	3	5	6
$S_8(x)$	0	3	5	8	6	9	C	7	D	A	E	4	1	F	B	2
$S_9(x)$	0	3	5	8	6	C	B	7	9	E	A	D	F	2	1	4
$S_{10}(x)$	0	3	5	8	6	A	F	4	E	D	9	2	1	7	C	B
$S_{11}(x)$	0	3	5	8	6	C	B	7	A	4	9	E	F	1	2	D
$S_{12}(x)$	7	C	E	9	2	1	5	F	B	6	D	0	4	8	A	3
$S_{13}(x)$	4	A	1	6	8	F	7	C	3	0	E	D	5	9	B	2
$S_{14}(x)$	2	F	C	1	5	6	A	D	E	8	3	4	0	B	9	7
$S_{15}(x)$	F	4	5	8	9	7	2	1	A	3	0	E	6	C	D	B

4. Differential Cryptanalysis

Differential cryptanalysis [7] is a cryptanalysis form related to cryptographic algorithm which takes the plaintext in blocks or stream. It will investigate how difference of change in input can affect the output change. Considering block cipher that refers to few techniques for depicting variations on the network of transformation, discovering the ciphertext exhibits some nonrandom characteristics and secret key taken back by using this nonrandom property. Differential cryptanalysis is normally a chosen-plaintext attack [8]; the cracker can retrieve ciphertext from known plaintext. Their analytical data rely on S- boxes used for encipher, so the attacker examines differentials (Δ_X, Δ_Y), where Δ_Y = S(X XOR Δ_X) XOR S(X) for each S-box.

Noticing the desired output difference between chosen or known plain cipher inputs recommends feasible values of key. If a differential of 1 => 1 (suggesting difference in LSB of input makes an output change in LSB) happens with probability of 4/256 (possible with the non-linear function in AES cipher) then for only 4 values (or 2 pairs) of inputs is that differential possible. For non-linear function where key is XOR'ed before assessment then the differential values are {2,3} and {4,5}. If the attacker sends the values {6, 7} and gets the correct output difference it means key is either 7 XOR K = 3, or 7 XOR K = 5, meaning the key K is either 3 or 5.In essence, for an n-bit non-linear function one must ideally search as close to $1/2^{(n-1)}$ to reach differential consistency. The differential attack needs more work to extract symmetric key. As the maximum chance of success in the Differential Distribution Table (DDT) [9] is 2^{-2}, the active substitution boxes measured from round-2 to round-4 are 3, 6 and 7 respectively.

4.1. Characteristics of Differential Cryptanalysis

For any ciphertext, it is mandatory to fix the difference in input proper so as to attain the attack success. An analysis of the algorithm's internals is taken up to track highly probable differences through the various encryption stages, mentioned as differential characteristic [10]. In this method, observing the desired output change between two chosen or unknown plaintext inputs suggests possible key values. Security on differential cryptanalysis is essential for contemporary block ciphers. The measure is generally validated by finding a equivalent of lower bound of active substitution boxes.

The plaintext will be differed by a few numbers of bits. It is launched as an adaptive chosen-plaintext attack; the attacker selects plaintext to be enciphered (but key is unknown) and then performs encryption on related plaintexts. In extract, for an n-bit non- linear function one must ideally inquire as close to $2^{-(n-1)}$ as possible to approach differential uniformity. When this is happened, the differential attack needs more work to acquire the key than brute forcing the key. The chance of success by exhaustive search (Brute-force attack) is 2^{-64} since key size is 64-bits. For the attack to be succeeded, the chance of differential characteristics [9] should be more than 2^{-64}.Differential propagation probability of an active substitution box at most = 2^{-2}.

Probability of differential characteristics of PRESENT algorithm = No. of active S-boxes * Probability of an active S-box. ---- (1)

Probability of differential characteristics of key dependent S-box = No. of active S-boxes * Probability of an active S-box * 2^{-4} ----- (2)

5. Results and Discussion

S-box plays an essential role in security assessment of cryptographic algorithm. PRESENT algorithm consists of single S-box with 16-hexadecimals.The total possibilities of all substitution boxes in a round is 2^4 x 2^4 = 2^8. In key dependent PRESENT algorithm [11], 16 S-boxes are used in a round. Henceforth the total possibilities of all S-boxes in a round are 2^4 x 2^4 x 2^4 = 2^{12}. In this paper, security analysis is concluded using active substitution boxes.

Table 4 explains the S-box chosen in a round for the 64-bits of plaintext with 80-bits of key. It is noted that different S-box was chosen in a round. The choice of different S-box in a round leads to the fullest extension of security by 2^4.

Table 4 Selection of S-Box

Plain text (64 bits)	Key (80 bits)	Round Number	S-Box chosen
5555ffffeeeea aaa	11111333335 555522222	9, 31	S_0
		14, 22	S_1
		3, 6, 23, 25, 28	S_2
		19, 20	S_3
		13, 15, 16, 24, 27	S_4
		1, 4	S_5
		8	S_6
		2, 10, 12, 30	S_7
		7	S_8
		18, 21, 26, 29	S_9
		5, 11	S_{12}
		17	S_{13}

Figure 4 shows simulated result of key dependent S-box obtained in Python. In this result, ciphertext is generated for the text input. The S box chosen for encryption is S_7.

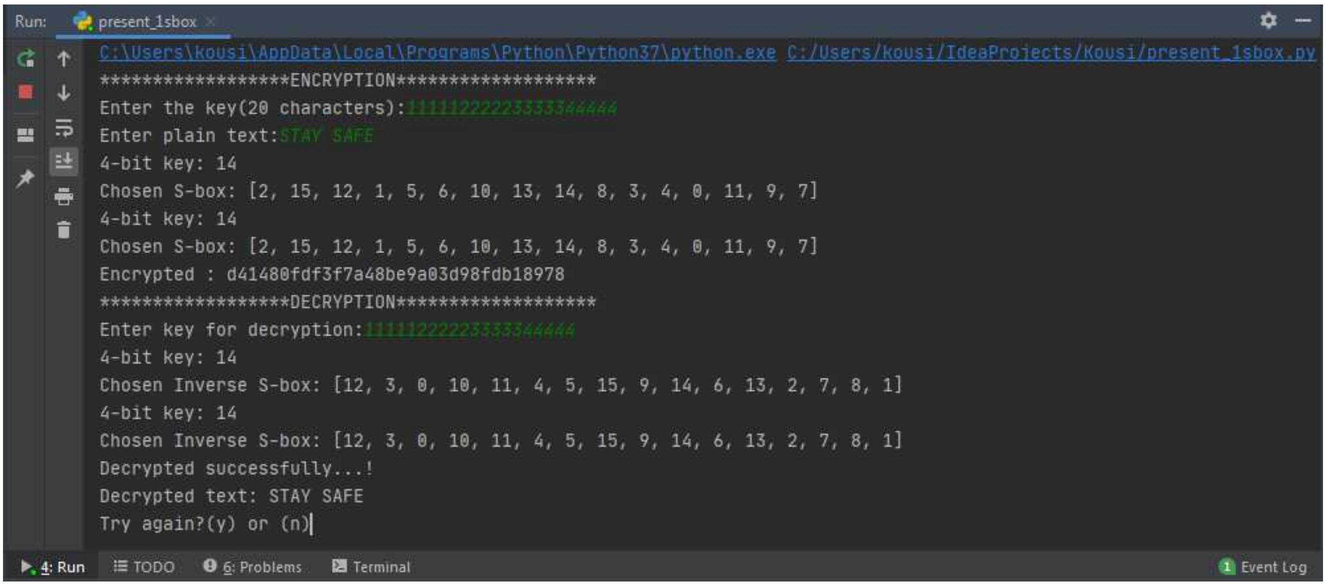

Figure 4 Encrypted and Decrypted Output

Figure 5 explains the substitution boxes activated in a round for key dependent technique. The active substitution boxes are the one with non-zero inputs. If the count of active substitution boxes [12] is high then the algorithm is more secure. Here 7 S-boxes are activated for 4 rounds.

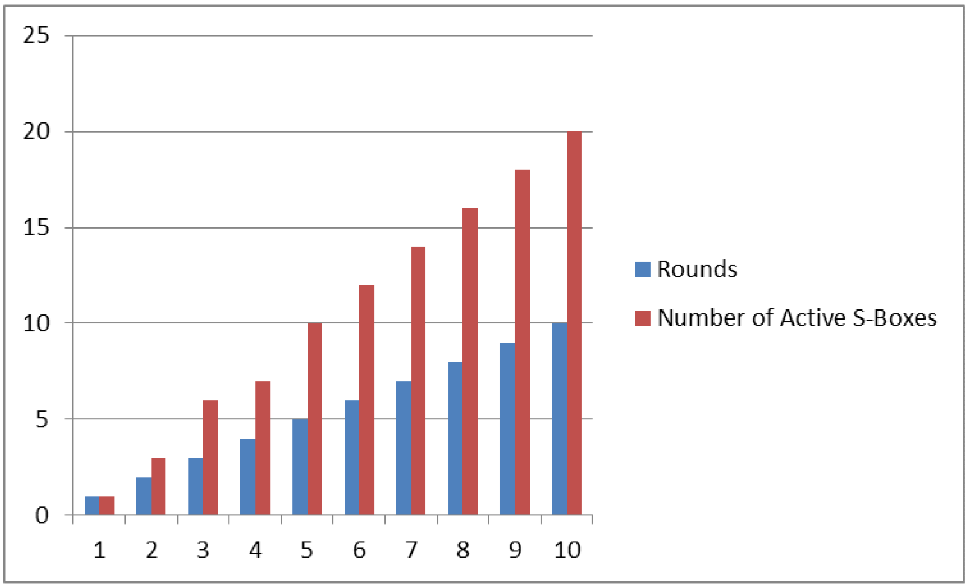

Figure 5 Active S-Boxes in each round

Table 5 shows the comparative assessment of probability of deciphering in PRESENT and key dependent S-box algorithm. It is noted that probability of success to decipher the ciphertext is reduced by 2^4.

Table 5 Comparison of Differential Cryptanalysis

Rounds	Number of Active S-Boxes	Differential Probability for PRESENT algorithm	Differential Probability for Key Dependent PRESENT algorithm
1	1	2^{-2}	2^{-8}
2	3	2^{-6}	2^{-24}
3	6	2^{-12}	2^{-48}
4	7	2^{-14}	2^{-56}
5	10	2^{-20}	2^{-80}
6	12	2^{-24}	2^{-96}
7	14	2^{-28}	2^{-112}
8	16	2^{-32}	2^{-128}
9	18	2^{-36}	2^{-144}
10	20	2^{-40}	2^{-160}

Figure 7 Original Image

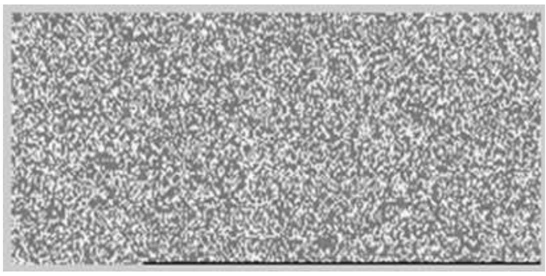

Figure 8 Encrypted Image

Figure 9 Decrypted Image

Figure 7 – 9 shows original image, Encrypted image and Decrypted image for key dependent technique. Since each round uses different S-boxes, the proposed technique is more secure [13] than the existed algorithm.

6. Conclusion

PRESENT algorithm with key dependent technique has been designed with 16 S-boxes. In each round, different S-boxes are chosen and the numbers of active substitution boxes are increased by 2^{-4}. Henceforth the differential probability of key dependent technique has been reduced by 2^4. The simulated results for text and image inputs are verified in Python programs. The performance of lightweight cryptographic system is validated by the area and security. Reversible logic gates can be used to design a hardware efficient S-box.

References

[1] M. Girija, Dr.P.Manickam, Dr.M.Ramaswami, "DIBPresent: A Dynamic Integer Based LightWeight Cryptography for Resource Constrained Devices", International Journal of Advanced Science and Technology vol. 29, No. 8s, **(2020)**, pp. 721-729.

[2] Jasvir Kaur, Brahmaleen Kaur Sidhu, "A Survey on Light-weight Block Ciphers for Wireless Sensor Network", International Journal of Advanced Research in Computer Science, vol. 8, No. 5, May – June **(2017)**.

[3] Kamsiah Mohamed, Mohd Nazran Mohammed Pauzi, Fakariah Hani Hj Mohd Ali, Suriyani Ariffin, Nurul Huda Nik Zulkipli, "Study of S-box properties in Block Cipher", IEEE International Conference on Computer, Communication and Control Technology, **(2014)**.

[4] Salim, Sufyan and Taha Alshaikhli, Imad Fakhri and Sulaiman, Alyaa Ghanim "Lightweight block cipher algorithms: review paper" International Journal of Enhanced Research in Science, Technology & Engineering, vol 5, **(2016)**. pp. 136-143.

[5] Jacob, G., A. Murugan, and I. Viola, "Towards the Generation of a Dynamic Key-Dependent S-Box to Enhance Security". IACR Cryptology ePrint Archive, **(2015)**. pp. 92.

[6] AlDabbagh, S.S.M. and Taha Alshaikhli, Imad Fakhri and Zaba, Muhammad Reza "Key dependent S-box in Light weight Block Cipher" Journal of Theoretical and Applied Information Technology, (**2014),** vol. 62 (2). pp. 554-559.

[7] Keshav Raj , Bharti Sharma , Neeraj Kumar , Dr. Dalveer Kaur, “Differential Cryptanalysis on S-DES” International Journal of Management & Information Technology ISSN: 2278-5612 vol 1, no 2, **(2012)** July, pp. 42-45.

[8] Amr Alasaad, Abdullah Alghafis, “Key-Dependent S-box Scheme for Enhancing the Security of Block Ciphers”, 2nd International Conference on Signal Processing and Information Security (ICSPIS), **(2019)**.

[9] Vikas Tiwari, Priyanka Garg, Ajeet Singh, “Differential Cryptanalysis on Block Ciphers: New Research Directions” International Journal of Computer Applications vol. 168 no.5, **(2017)** June pp. 1 – 7.

[10] Sufyan Salim Mahmood AlDabbagh and Imad Al Shaikhli,” Lightweight Block Ciphers: a Comparative Study”, in Journal of Advanced Computer Science and Technology Research vol.2 No.4 **(2012)**, November pp. 159-165.

[11] Vikas Tiwari, Ajeet Singh, Appala Naidu Tentu “Differential cryptanalysis on DES cryptosystem up to eight rounds” International Journal of Information Privacy, Security and Integrity, vol.4 No.1, (**2019**), pp.1 – 29.

[12] M Sajadieh, A Mirzaei, H Mala, V Rijmen, “A new counting method to bound the number of active S-boxes in Rijndael and 3D”, Designs, Codes and Cryptography 83 (2), Springer, **(2017)**, pp. 327-343.

[13] M.Sri Lakshmi, V.Srikanth, "A Study on Light-Weight Cryptography Algorithms for Data Security in IOT", International Journal of Engineering and Technology, vol. 7(2.7), **(2018),** pp. 887-890.