International Conference on Advances in Materials, Computing and Communication Technologies
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Hardware Efficient LED for IoT Applications Using B-RAM

B. Charu Manjari, M S Nivetha Nandini, K Susmitha, Dr. A. Babu Karuppiah, Mr. R. Rajaraja
Velammal College of Engineering and Technology Viraganoor, Madurai-625009, Tamil Nadu, India

ABSTRACT

This paper analyses to implement a hardware efficient light weight encryption algorithm based on Light Encryption Device (LED). The hardware efficiency of a LED is mainly determined by the implementation of the Substitute Cell and the Mix Columns operation. In order to increase the speed, these two round operations are combined into a single step called Transformation Box (T-Box). To implement the designed LED algorithm, we use an iterative architecture so that the hardware elements can be reused for every round operation. Further Block RAMs (BRAMs) are utilized for reducing area utilization. We use 64 bit plain text and 128 bit key size to get 64 bit cipher text which is targeted to Spartan 3 FPGA.

Index terms: lightweight encryption algorithm, increase the speed, Block RAMs (BRAMs), reducing area utilization, Spartan 3 FPGA.

I. INTRODUCTION

LED block cipher : The LED cipher is a 64-bit block cipher with two primary instances taking 64- and 128-bit keys . They have 4 steps in one round. They are add constant , substitute cells , shift rows and mix columns serial. LED block cipher cryptography: The technique used to precisely evaluate the number of rounds to ensure proper security. LightWeight Cryptography is an encryption method used to lower computational complexity. Cryptography provides different algorithm for securing and authenticating the transmission of information over the channels. RFID and sensor nodes contain sensitive information and confidential information such device are used in many application but these miniature device are not possible to run in traditional cryptography which require large memory and greater power. In order to satisfy these constraint Lightweight cryptography is used.

II. LED ALGORITHM

Light Weight Encryption Device (LED) block cipher is a 64 bit block cipher that uses cryptographic key sizes from 64 bits to 128 bits. For 32 rounds of operation, 64 bit key size is used and for 48 rounds of operation, 128 bit key size is used. This work deals with 128 bit key size of LED algorithm due to security concern . Table.1 shows the number of rounds of operation involved for corresponding key size. Since 128 bit key size provides more security, this work deals with 128 bit key size of LED algorithm.

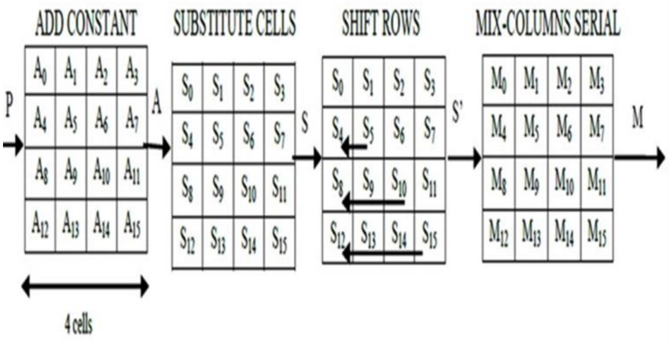


Fig. 1.Four round operation

Add round key

The Add Round Key operation combines nibbles of K1 or K2 with the state, corresponding array positioning, using bitwise Exclusive-OR.In ADD Constants a round key is added to the state by a simple bit-wise XOR operation. There is no key schedule, or rather this is the sum total of the key schedule, and the arrays K1 and appropriate K2 are alternatively used without modification . Add Round Key is equivalent to a bitwise XOR and therefore in decryption it is an inverse of itself.

Add constants

A round constant is defined as follows. At each round, six bits (rc5,rc4,rc3, rc2, rc1, rc0) are shifted one position to the left with the new value to rc0 being computed as rc5⊕rc4⊕1.The round constants are combined with the state, respecting array positioning, using bitwise EXclusive-OR. The values of the constants for each round of LED encryption are given in Table.1 which are encoded to Hexadecimals values for each round, with being the least significant bit. The round constants are combined with the state, respecting array positioning, using bitwise EXclusive-OR. The values of the constants for each round of LED encryption are given in Table.1 which are encoded to Hexadecimals values for each round, with being the least significant bit .

Table.1 Round constants (LED encryption)

Rounds	Constants
1-12	01,03,07,0F,1F,3E,3D,3B,37 ,2F,1E,3C
13-24	39,33,27,0E,1D,3A,35,2B,1 6,2C,18,30
25-36	21,02,05,0B,17,2E,1C,38,31 ,23,06,0D
37-48	1B,36,2D,1A,34,29,12,24,0 8,11,22,04

Substitute Cells

The substitute nibble transformation is a non-linear nibble substitution that operates independently on each nibble of the state using a substitution table (S-Box). The S-Box which is invertible is 4-bit to 4-bit S-Box. Computation can be avoided by storing the pre-calculated values in memory.

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

Table.2 S-Box

In LED encryption each element in the state array is replaced by the element generated after using the S-Box. The action of the S-Box in Hexadecimal notation is given in Table.2

Shift Row

Row i of the array state is rotated i cell positions to the left, for i = 0, 1, 2, 3.

The shift row transformation consist of

(i) not shifting the first row of the state array;

(ii) The second row is circularly shifting by one byte to the left;

(iii)The third row is circularly shifting by two bytes to the left; and

(iv)The last row is circularly shifted by three bytes to the left

Mix Columns Serial

Each column of the array state is viewed as a column vector and replaced by the column vector that results after post-multiplying the vector by the matrix M.The final value of the state provides the ciphertext with nibbles of the "array" being unpacked in the obvious way. Test vectors for LED are provided in the Appendix.

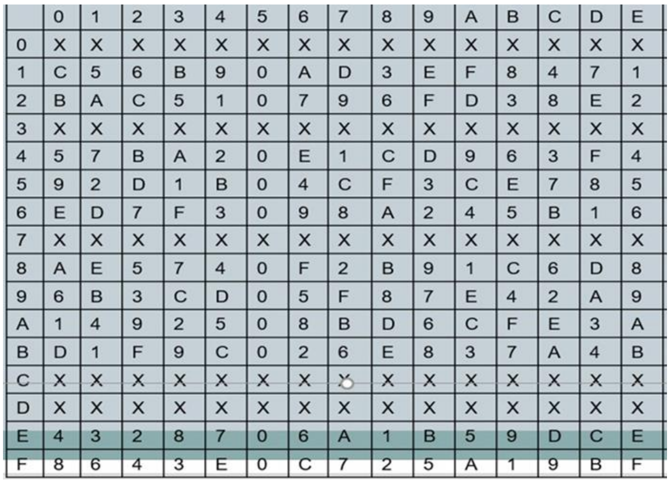
III. PROPOSED METHODOLOGY

The hardware implementation of LED block cipher by Jian Guo, Thomas Peyrin, Axel Poschmann and Matt Robshaw has been proposed in 2011 and it is implemented using Mentor graphics. Their work is based on serial hardware architecture of LED algorithm.It requires four rounds of operation i.e. Add Constant, Substitute Cell, Shift Rows, Mix Columns Serial. In this work the Mix Columns Serial operation of LED algorithm is implemented using an efficient GF ( ) multiplier.

Transformation Box (T - Box)

T-BOX is manually calculated.

The possible inputs of two operations (substitute cells ,and mix column serial) are calculated and updated in T-BOX.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1
2	B	A	C	5	1	0	7	9	6	F	D	3	8	E	2
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	5	7	B	A	2	0	E	1	C	D	9	6	3	F	4
5	9	2	D	1	B	0	4	C	F	3	C	E	7	8	5
6	E	D	7	F	3	0	9	8	A	2	4	5	B	1	6
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	A	E	5	7	4	0	F	2	B	9	1	C	6	D	8
9	6	B	3	C	D	0	5	F	8	7	E	4	2	A	9
A	1	4	9	2	5	0	8	B	D	6	C	F	E	3	A
B	D	1	F	9	C	0	2	6	E	8	3	7	A	4	B
C	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
D	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
E	4	3	2	8	7	0	6	A	1	B	5	9	D	C	E
F	8	6	4	3	E	0	C	7	2	5	A	1	9	B	F

Table. 3 Manually Calculated T-Box

By using T-BOX ,we get the predicted input values for the corresponding output.

Thus increasing the speed of the process.

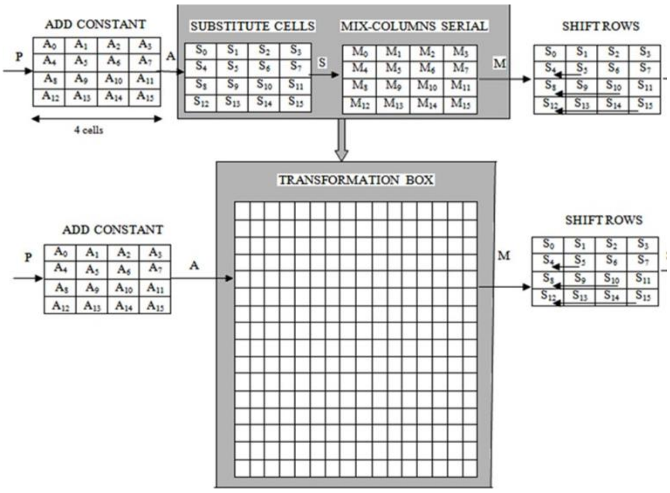


Fig. 2 The proposed method

This method of deriving Transformation box will result in reduced computations and enhanced speed but the slice utilization of FPGA is increased. This area (slice) utilization drawback can be overcome by configuring the BRAMs of FPGA to store the values of T-Box.

IV. SIMULATION AND RESULT

The LED encryption algorithm is designed using verilog HDL and simulated using Xilinx ISim simulator. The algorithm is verified using appropriate test inputs as shown below (represented in Hexadecimal). The encryption module is simulated with a 64 bit input plain text and 128 bit key.

Plain text: 0123456789ABCDEF

Key:0123456789ABCDEF0123456789AB CDEF

After 48 rounds of operation the cipher text obtained is, **Cipher text: 3131C231205C3664**

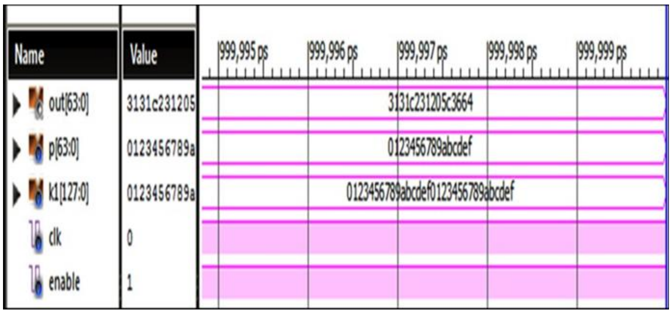
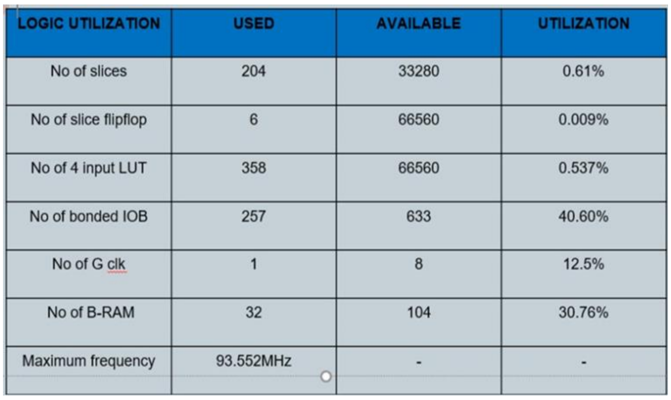


Fig. 3 Simulation(LED Encryption)

Fig.3 . shows the simulation result of LED encryption (48 rounds). It shows the 64 bits plain text and k1 shows the 128 bits key size. Always at the positive edge clock and the enable signal at 1, the 64 bit generated encrypted output is produced which is shown as out.



LOGIC UTILIZATION	USED	AVAILABLE	UTILIZATION
No of slices	204	33280	0.61%
No of slice flipflop	6	66560	0.009%
No of 4 input LUT	358	66560	0.537%
No of bonded IOB	257	633	40.60%
No of G clk	1	8	12.5%
No of B-RAM	32	104	30.76%
Maximum frequency	93.552MHz	-	-

It is found that a single Transformation Box occupies totally 1024 bits (256X4) and since there are 48 rounds of operation it utilizes 49152 bits i.e., totally 48 Kilo Bytes of memory. From the Transformation Box it is found that some of the rows are useless and their corresponding row addresses do not take part in the MDS matrix.

Hence in Encryption Transformation Box there are 5 rows corresponding to the row addresses of 1, 3, 7, C and D are completely eliminated and this leads to the memory usage of only 33792 rather than 49152 bits of LED Encryption algorithm. The Transformation Box for LED Encryption with reduced memory usage is shown in Fig.3

In this paper work,The transformation box has been successfully derived for encryption. The T-Box based LED implementation occupies limited area when compared to the LED implementation using GF multiplier. The LED encryption using T-Box occupies 204 slices, 6 flip flops and 358 four input LUTs. The speed of LED encryption is enhanced to 93.552 MHz from 79.012 MHz.

V. CONCLUSION

This work is mainly intended to reduce area utilization and improve the speed of LED encryption. The other aspects of algorithm such as reducing the number of rounds by maintaining the provable security and key scheduling are not considered here. These aspects may be addressed for future work. Further improvements in speed can be obtained by using pipelined architecture.

VI. REFERENCES

[1]. Li Yue, Li Wei, Cao Yanqin and Le Jiajin, "Performance analysis of several lightweight block cipher[J]“.

[2]. Wang Chenxu, Han Liang, Yu Mingyan and Wang Jinxiang, "Implementation of a secure cryptographic algorithm for RFID Tags[J]“.

[3]. Wang Ya, Wei Guoheng and Wei Wei, "Research on classification model of lightweight encryption algorithm for RFID applications[J]“.

[4]. A Kumar, K Gopal and A Aggarwal, "A novel lightweight key management scheme for RFID-sensor integrated hierarchical MANET based on internet of things[J]“.

[5]. D Dinu, Y L Corre, D Khovratovich et al., Triathlon of lightweight block ciphers for the internet of things[R], IACR ePrint archive, 2015.

[6]. A K. Manjulata, "Survey on lightweight primitives and protocols for RFID in wireless sensor networks[J]", 2014.

[7]. M Tausif, J Ferzund, S Jabbar et al., "Towards Designing Efficient Lightweight Ciphers for Internet of Things[J]", 2017.