ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Area Delay Efficient Encryption Design Using LED for Embedded Security

Honey Devaraj[1], Sharon Mathew[2]

P.G. Student, Department of Electronics and Communication Engineering, Indira Gandhi institute of Engineering and Technology for women, Nellikuzhy, Kerala, India[1]

Associate Professor, Department of Electronics and Communication Engineering, Indira Gandhi institute of Engineering and Technology for women, Nellikuzhy, Kerala, India[2]

ABSTRACT: In this project report here present a new block cipher LED. While dedicated to compact hardware implementation, and offering the smallest silicon footprint among comparable block cipher, the cipher has been designed to simultaneously tackle 3 additional goals. First here explore the role of an ultra- light(in fact non-existent) key schedule. Secondly consider the resistance of ciphers, and LED in particular, to related- key attacks : Here it is able to derive simple yet interesting AES-like security proofs for LED regarding related- or single-key attacks. And third, here provide a block cipher that is very compact in hardware, aim to maintain a reasonable performance profile for software implementation.

KEYWORDS: Lightweight, Block cipher, LED, Embedded security, Encryption.

I. INTRODUCTION

With the establishment of the AES the need for new block ciphers has been greatly diminished; for almost all block cipher applications the AES is an excellent and preferred choice. However, implementation advances, the AES is not suitable for extremely constrained environments such as RFID tags and sensor networks. In recent years, many compact algorithms like PRESENT, CLEFIA, SEA,TEA, LED,ZORRO, Hummingbird, and KANTAN have made the mark to be used as lightweight crypto engines. Research is a process of arriving at an appropriate solution to a problem through a systematic approach through lightweight encryption system for embedded security.

Over past years many new cryptographic primitives have been proposed for use in RFID tag deployments, sensor networks, and other applications characterized by highly-constrained devices. The pervasive deployment of tiny computational devices brings with it many interesting, and potentially difficult, security issues. Chief among recent developments has been the evolution of lightweight block ciphers where an accumulation of advances in algorithm design, together with an increased awareness of the likely application, has helped provide important developments. In this project work, therefore, return to the design of lightweight block ciphers and describe Light Encryption Design, LED.

The cipher has been designed to simultaneously tackle three additional goals. First, here explore the role of an ultra-light (in fact non-existent) key schedule. Secondly consider the resistance of ciphers, and LED in particular, to related-key attacks: It is able to derive simple yet interesting AES-like security proofs for LED regarding related- or single-key attacks. And third, here provide a block cipher that is very compact in hardware, it is aim to maintain a reasonable performance profile for software implementation. Practically all modern block cipher proposals have reasonable security arguments. Here it is able to provide a more complete security treatment than is usual. In particular, related key attacks are often dismissed from consideration for the application areas that typically use such constrained

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

devices. So having provable levels of resistance to such attacks would be a bonus and might help confusion developing in the cryptographic literature.

This project work is organized as follows: In section II Related works are shown, section III deals with conventional encryption system, section IV deals with proposed design, section V describes design approach and specification, section VI describes security analysis. section VII deals with key schedule. Section VIII analyses results and section IX concludes.

II.RELATED WORK

A.PRESENT—AN ULTRA LIGHTWEIGHT BLOCK CIPHER

In this project work, here describe an ultra-lightweight block cipher, present. Both security and hardware efficiency have been equally important during the design of the cipher and at 1570 GE, the hardware requirements for present are competitive with today's leading compact stream ciphers. In this work, here introduce a new hardware-optimized block cipher that has been carefully designed with area and power constraints uppermost in our mind. Yet, at the same time, there also tried to avoid a compromise in security. In achieving this first looked back at the pioneering work embodied in the DES and complemented this with features from the AES finalist candidate Serpent which demonstrated excellent performance in hardware.

PRESENT is an example of an SP-network and consists of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. Given the applications it recommends the version with 80-bit keys. PRESENT is an example of an SP-network and consists of 31 rounds. Each of the 31 rounds consists of anxor operation to introduce a round key .The S-box used in present is a 4-bit to 4-bit S-box. The action of this box in hexadecimal notation.Fig1. below shows the algorithmic description of PRESENT.

```
generateRoundKeys()
for i = 1 to 31 do
addRoundKey(state,Ki)
sBoxLayer(state)
pLayer(state)
end for
addRoundKey(state,K32)
```

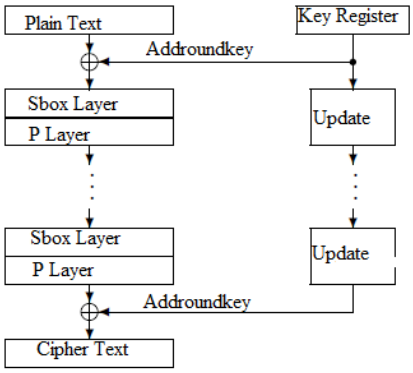

Fig 1: Algorithmic description of PRESENT

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

B. BIT PERMUTATION INSTRUCTION FOR ACCELERATING SOFTWARECRYPTOGRAPHY.

Permutation is widely used in cryptographic algorithms. In this project, two novel permutation instructions that can perform arbitrary permutations of nbits efficiently. There has been limited past work on general-purpose permutation instructions. One PERMUTE instruction can generate in a single cycle any permutation of four 16-bit subwords, including permutations with repetition of elements. The MIX instruction can combine the contents of two registers. If we extend it to handle 1-bit subwords, it can perform many regular permutations efficiently. The basic idea of the GRP instruction is to divide the bits in the source R1 into two groups according to the bits in R2. For each bit in R1, we check the corresponding bit in R2. If the bit in R2 is 0, we put this bit in R1 into the first group. Otherwise put this bit in R1 into the second group. During this process do not change the relative positions of bits in the same group.Finally, putting the first group to the left of the second group, there get the result value in R3. From the position of two groups, here call the first the left group, and the second the right group. Figure 2. shows how the GRP instruction works on 8-bit systems.

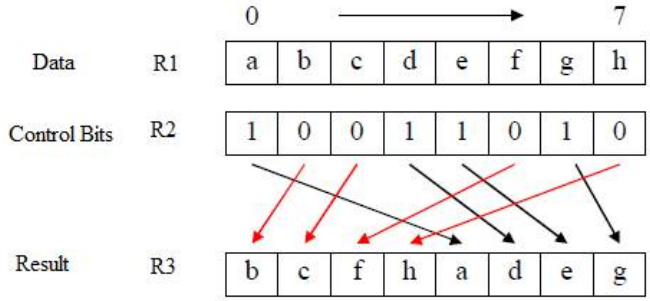

Fig 2: GRP instruction on 8 bit system

In this work, every time here generate a new arrangement, its numMIS value decreases by half. Also know that the maximum numMIS value of the final arrangement is its length n. It implies that we need at most lg(n) steps to reduce the numMIS value of arrangements from n to 1. Thus, the number of iterations does not exceed lg(n) and the number of valid entries in the returned array aCdoes not exceed lg(n). For each entry in aC, I generate one GRP instruction. The total number of GRP instructions does not exceed lg(n). When performing the permutation P, it start from the initial arrangement whose numMISvalue is 1. Using the control bits kept in array aCfrom entry (num_instr– 1) to entry 0, here generate the temporary arrangements in the reverse order, and achieve the final arrangement after the last GRP instruction. Table I shows what happens on the example as mentioned above, an 8-bit permutation: (7,6,5,4,3,2,0,1).

Table I: GRP instruction sequence for an 8 bit permutation

Iteration	1	2	3
P	(7,6,5,4,3,2,0,1)	(3, 7, 2, 6, 0, 1, 5, 4)	(0, 1, 3, 5, 7, 2, 4, 6)
MISes in P	(7)(6)(5)(4)(3)(2)(0,1)	(3,7)(2,6)(0,1,5)(4)	(0, 1, 3, 5, 7)(2, 4, 6)
After Alg. 1, step 1	(7,3)(6,2)(5,0,1)(4)	(3,7, 0,1,5)(2,6,4)	(0, 1, 3, 5, 7, 2, 4, 6)
After Alg. 1, step 2	Q = (3,7)(2,6)(0,1,5)(4)	Q=(0, 1, 3, 5, 7)(2, 4, 6)	(0, 1, 2, 3, 4, 5, 6, 7)
After Alg. 1, step 3	c = 10101100	c = 11010010	c = 00101010

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

C.THE LED BLOCK CIPHER IN CRYPTOGRAPHIC HARDWARE ANDEMBEDDED SYSTEM

In this work, here present a new block cipher LED. The cipher has been designed to simultaneously tackle three additional goals. First, here explore the role of an ultra-light (in fact non-existent) key schedule. Secondly consider the resistance of ciphers, and LED in particular, to related-key attacks: It is able to derive simple yet interesting AES-like security proofs for LED regarding related- or single-key attacks. And third, here provides a block cipher that is very compact in hardware. The LED cipher is a 64-bit block cipher with two primary instances taking 64- and 128-bit keys. The cipher state is conceptually arranged in a (4×4) grid. The LED block cipher is simple to analyze and this allows us to precisely evaluate the necessary number of rounds to ensure proper security. This scheme is meant to be resistant to classical attacks, but also to the type of related-key attacks that have been effective against AES-256 and other ciphers. Here even study the security of LED in a hash function setting. In other words, it will consider attackers that have full access to the key(s) and try to distinguish the fixed permutations from randomly chosen ones. While this analysis provides additional confidence in the security of LED, it is not our intent to propose a hash function construction.

III. CONVENTIONAL ENCRYPTION SYSTEM PRESENT

Lightweight cryptography is an interesting field that strikes the perfect balance in providing security, higher throughput, low-power consumption, and compactness. In recent years, many compact algorithms like PRESENT, CLEFIA, SEA, TEA, LED, ZORRO, Hummingbird, and KANTAN have made the mark to be used as lightweight crypto engines. In this project work, first here present the design of a new lightweight compact encryption system based on bit permutation instruction group operation (GRP), which is widely studied and extensively researched. Using the S-box of PRESENT, here added the confusion property for GRP. A hybrid cryptosystem, which consists of GRP and S-box of PRESENT, is designed and implemented on a 32-bit processor. This fusion has resulted in a lightweight cipher that is the most compact implementation, till now, in terms of memory requirement.

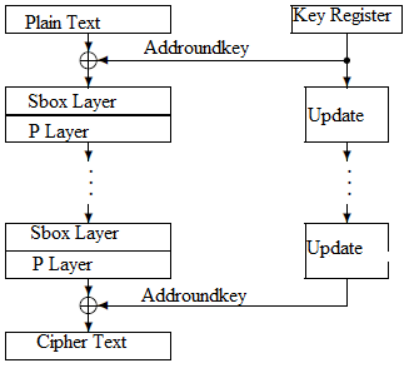

Fig 3: Algorithmic description of PRESENT

The aim of this work is to describe the results of a design of a compact cipher with adequate security for applications like pervasive computing. In this research work focuses on a compact hardware implementation of the cipher. Bit permutation instructions are efficient in such kind of implementations and they are complex in nature which gives them an edge in cryptographic environment. Bit permutations are popularly known to be used in permutation block known as diffusion property.PRESENT structure is Substitution and Permutation (SP) design which consists of S and P layer and consists of 31 rounds. In this work, separate analysis is done for S-box as well as for P-box. The algorithm of PRESENT is shown in Fig 3. with 64 bit block size and key size of 128 bits.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

A.GRP Implementation

GRP algorithm can generate the different keys at different rounds from a given integer sequence. Fig.4 indicates the basic GRP encryption operations in terms of AND, OR and NOT gates. Key register generates the key according to GRP algorithm , that is based on the user defined integer sequence and that key is applied as code word to each of the permutations to do the encryption.For example, for P = 64, C value will be 1, for second stage where P = 32, C value will be 2 and for last stage P = 1, C value will be 64 which means we will be having 64 combinations of single pair.

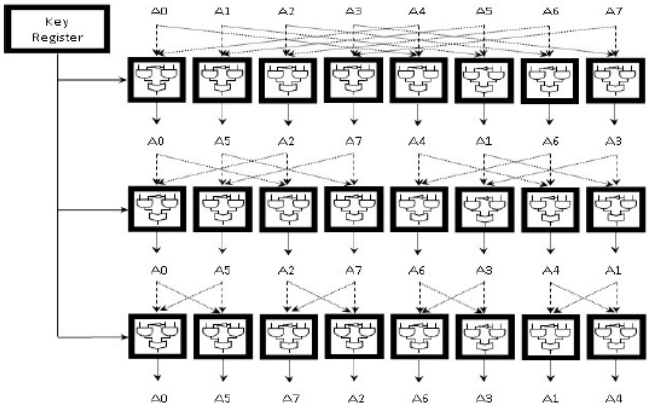

Fig 4: GRP implementing encryption for 8 bit.

IV. PROPOSED ENCRYPTION SYSTEM LED

In this project here presents a new block cipher LED. While dedicated to compact hardware implementation, and offering the smallest silicon footprint among comparable block ciphers, the cipher has been designed to simultaneously tackle three additional goals. First, here explore the role of an ultra-light (in fact non-existent) key schedule. Secondly consider the resistance of ciphers, and LED in particular, to related-key attacks: it is able to derive simple yet interesting AES-like security proofs for LED regarding related- or single-key attacks. And third, here provide a block cipher that is very compact in hardware, it aims to maintain a reasonable performance profile for software implementation. Over past years many new cryptographic primitives have been proposed for use in RFID tag deployments, sensor networks, and other applications characterized by highly-constrained devices. The pervasive deployment of tiny computational devices brings with it many interesting, and potentially difficult, security issues.

Chief among recent developments has been the evolution of lightweight block ciphers where an accumulation of advances in algorithm design, together with an increased awareness of the likely application, has helped provide important developments. To some commentators the need for yet another lightweight block cipher proposal will be open to question. However, in addition to the fact that many other works present some weaknesses, it feel there is still more to be said on the subject and observe that it is in the “second generation” of work that designers might learn from the progress, and omissions, of “first generation” work. And while new proposals might only slightly improve on successful initial proposals in terms of a single metric, e.g. area, they might, at the same time, overcome other important security and performance limitations. In this work, therefore, it return to the design of lightweight block ciphers and describe Light Encryption Device, LED.

During our design, several key observations were uppermost in our mind. Practically all modern block cipher proposals have reasonable security arguments; but few offer much beyond (potentially thorough) ad hoc analysis. Here it is able to provide a more complete security treatment than is usual. In particular, related key attacks are often

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

dismissed from consideration for the application areas that typically use such constrained devices, e.g. RFID tags. In practice this is often perfectly reasonable. However, researchers will continue to derive cryptanalytic results in the related-key model and there has been some research on how to modify or strengthen key schedules. So having provable levels of resistance to such attacks would be a bonus and might help confusion developing in the cryptographic literature.

In addition, it is more focused on the performance of the algorithm on the tag. However, there can be constraints when an algorithm is also going to be implemented in software. This is something that has already been discussed with the design of KLEIN and in the design of LED ,it have aimed at very compact hardware implementation while maintaining some software friendly features.

The new block cipher is based on AES-like design principles and this allows us to derive very simple bounds on the number of active Sboxes during a block cipher encryption. Since the key schedule is very simple, this analysis can be done in a related-key model as well; i.e. bounds apply even when an attacker tries to mount a related-key attack. And while AES-based approaches are well-suited to software, they don't always provide the lightest implementation in hardware. But using techniques presented it is able to resolve this conflict.

V.DESIGN APPROACH AND SPECIFICATION

Like so much in today's symmetric cryptography, an AES-like design appears to be the ideal starting point for a clean and secure design. The design of LED will inevitably have many parallels with this established approach, and features such as Sboxes, ShiftRows, and will all feature and take their familiar roles.

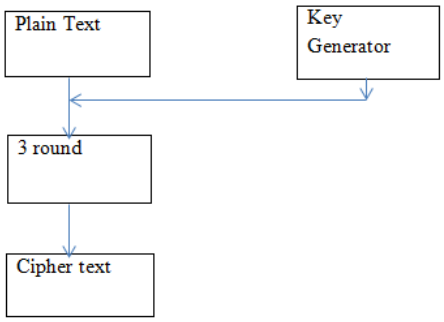

Fig 5:Block diagram of LED

For the key schedule here uses to do-away with the "schedule", i.e. the user provided key is used repeatedly as is. As well as giving obvious advantages in hardware implementation, it allows for simple proofs to be made for the security of the scheme even in the most challenging attack model of related keys. At first sight the re-use of the encryption key without variation appears dangerous, certainly to those familiar with slide attacks and some of their advanced variants. But note that such a simple key schedule is not without precedent though the treatment here is more complete than previously. The LED cipher is described is a 64-bit block cipher with two primary instances taking 64- and 128-bit keys. The cipher state is conceptually arranged in a (4×4) grid.

Sboxes: LED cipher re-uses the present Sbox which has been adopted in many lightweight cryptographic algorithms. The action of this box in hexadecimal notation is given by the following table.

MixColumnsSerial. Here re-use the tactic adopted in to define an MDS matrix for linear diffusion that is suitable for compact serial implementation. The MixColumnsSerial layer can be viewed as four applications of a hardware-friendly matrix A with the net result being equivalent to using the MDS matrix M .

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

The operation addRoundKey(state,Ki) combines nibbles of subkeyKiwith the state, respecting array positioning, using bitwise exclusive-or. There is no key schedule, or rather this is the sum total of the key schedule, and the arrays K1 and, where appropriate, K2 are repeatedly used without modification. Encryption is described using the previously mentioned addRoundKey(state,Ki) and a second operation, step(state). This is illustrated in Figure 6.

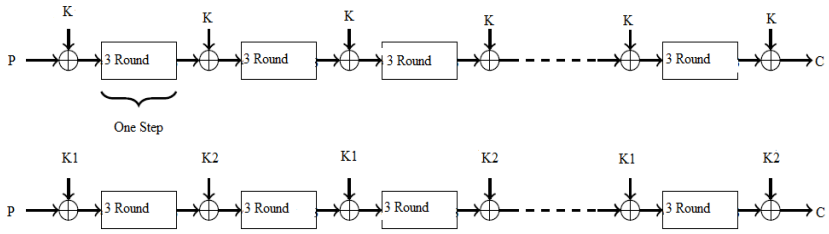

Fig 6: Use of key array in LED

The number of steps during encryption depends on whether there are one or two key arrays. The operation consists of four rounds of encryption of the cipher state. Each of these four rounds uses, in sequence, the operations Add Constants, Sub Cells, Shift Rows, and Mix Columns Serialas illustrated in Figure 6. Fig 7.shows the working in the 3 round .Also which are described detailed below.

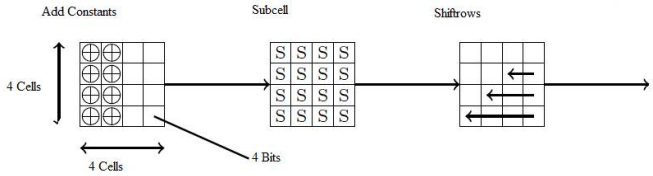

Fig 7: Single round of LED

AddConstants. A round constant is defined as follows. At each round, the 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 six bits are initialized to zero, and updated before use in a given round. The constant, when used in a given round, is arranged into an array as follows:

$$\begin{bmatrix} 0 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 1 & (rc_2||rc_1||rc_0) & 0 & 0 \\ 2 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 3 & (rc_2||rc_1||rc_0) & 0 & 0 \end{bmatrix}$$

Fig 8.Addconstant

The round constants are combined with the state, respecting array positioning, using bitwise exclusive-or. In sub Cells ach nibble in the array state is replaced by the nibble generated after using the present Sbox .Shift Rowis of the array state is rotated *i* cell positions to the left, fori= 0, 1, 2, 3. In 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.

VI. SECURITY ANALYSIS

The LED block cipher is simple to analyze and this allows to precisely evaluate the necessary number of rounds to ensure proper security. This scheme is meant to be resistant to classical attacks, but also to the type of

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

related-key attacks that have been effective against AES-256 and other ciphers. Here it will even study the security of LED in a hash function setting, i.e. when it is used in a Davies-Meyer or similar construction with a compression function based on a block cipher. In other words, it will consider attackers that have full access to the key(s) and try to distinguish the fixed permutations from randomly chosen ones. While this analysis provides additional confidence in the security of LED, it is not our intent to propose a hash function construction. In this project , here chose a conservative number of rounds for LED. For example, when using a 64-bit key array we use 32 AES-like rounds that are grouped as eight "big" addkey/ apply-permutation steps that are each composed of four AES-like rounds. Further, the security margins are even more conservative if one definitively disregards related-key attacks; as will be seen with the following proofs.

VII.THE KEY SCHEDULE

The LED key schedule has been chosen for its simplicity and security. Because itis very simple to analyze, it allows us to directly derive a bound on the minimal number of active Sboxes, even in the scenario of related-key attacks. The idea is to first compute a bound on the number of active big steps (each composed of 4 AES-like rounds). Then, using the well known 3-round proofs for the AES, one can show that one active big step will contain at least 25 active Sboxes. Note that this bound is tight as we know 3-round differential paths containing exactly this number of active Sboxes. When not considering related-key attacks, here directly obtain that any differential path for LED will contain at least [r/4].25 active Sboxes. For related-key attacks, here it have to distinguish between the different key-size versions.

A. 64-BIT KEY VERSION

If here assume that differences are inserted in the key input, then every subkeyK1 in the 64-bit key variant of LED will be active. Therefore, one can easily see that it is impossible to force two consecutive nonactivebig steps and ensured that for every two big steps at least one is active. Overall, this shows that any related-key differential path contains at least [r/8].25 active S boxes.

B. 128-BIT KEY VERSION

If here assume that differences are inserted in the key input, then there have to separate two cases. If the two independent parts K1and K2 composing the key both contain a difference, then it have to end up with exactly the same reasoning as for the 64-bit key variant: at least [r/8].25 active S boxes will be active. If only one of the two independent parts composing the key contains a difference, then subkeys with and without differences are alternatively incorporated after each big step. For LED-128, since here using two independent key parts one can peel off the first and last key addition (which is always the first key part K1). Thus, an attacker can remove one big step on each side of the cipher, for a total of 8 rounds, with a complexity of 264 tries on K1. This partially explains why the versions of LED using two independent key parts have 16 more rounds than for LED-64.

VIII.RESULTS AND DISCUSSION

The symmetric lightweight efficient LED algorithm is designed using VHDL and simulated using modelsim simulator and synthesized using the Xilinx synthesizer and targeted in FPGA device Spartan 6. The modelsim simulation results of LED block cipher Encryption and Decryption is shown in Figure 9 and Figure 10 respectively. In this project work FPGA device SPARTAN 6 is targeted.Spartan-6 devices are the most cost-optimized FPGAs, offering industry leading connectivity features such as high logic small form-factor packaging, and a diverse number of supported I/O protocols. Built on 45nm technology, devices are ideally suited for a range of advanced bridging applications found in automotive infotainment, consumer, and industrial automation. The encryption is verified using

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

the input which is shown below (in Hexadecimal representation). Table II showscomparison of area and delay in conventional and proposed design.
Plain text :0123456789abcdef
KEY :0123456789abcdef
Cipher Text:C97c947a09019e68

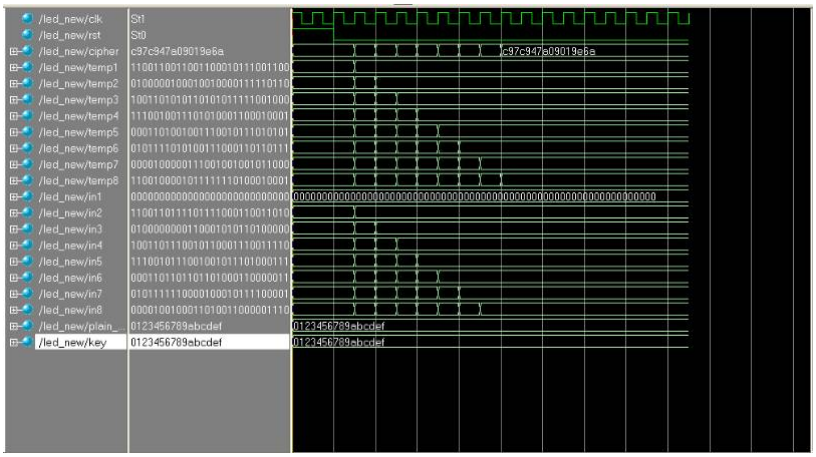

Fig 9: Simulation of encryption module

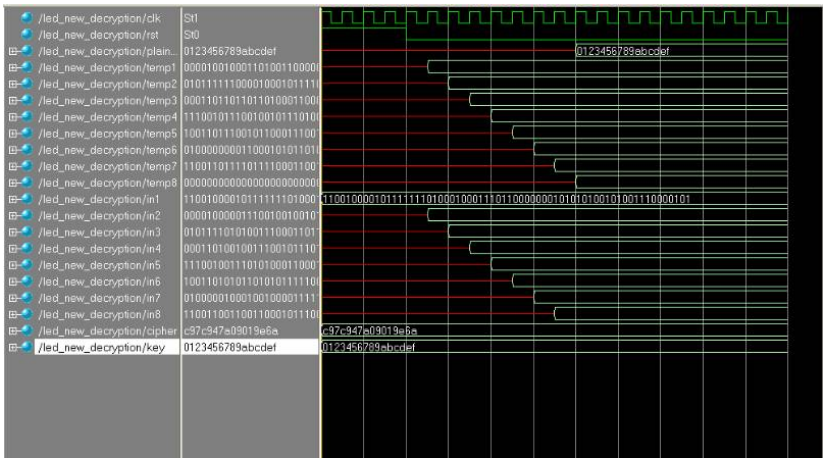

Fig 10:Simulation of decryption module

Table II. Comparison of area and delay in encryption system

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

	Area(No . of slices)	Delay(ns)
Conventional PRESENT	1630 out of 1728	7.99
Modified LED	218 out of 1728	5.69

IX.FUTURE SCOPE AND CONCLUSION

The new block cipher is based on AES-like design principles and this allows us to derive very simple bounds on the number of active S boxes during a block cipher encryption. In this project work , here presents the encryption and decryption in LED. In previous wok PRESENT, its area and delay is higher and for improving , here chooses LED so that its area and delay reduces. This wok can b further modified to reduce power consumption. In this project work , here presented the block cipher LED. Clearly, given its novelty, the cipher should not be used in applications until there has been sufficient independent analysis. Nevertheless, it hopes that the design is of some interest and it have focused our attention on what seem to be the neglected areas of key schedule design and protection against related-key attacks. Furthermore, here it done so while working in one of the more challenging design spaces— that of constrained hardware implementation—and have proposed one of the smallest block ciphers in the literature (for comparable choices of parameters) while striving to maintain a competitive performance in software Cryptography algorithm are omnipresent in modern communication in which the information security, such as confidentiality of communication or reliable authentication is absolute necessities. Efficient implementation techniques are necessary in order to provide high-security cryptographic algorithms. Hence in this work, encryption and decryption of Lightweight LED block cipher was successfully simulated using modelsim simulator and synthesized using the Xilinx synthesizer and targeted in FPGA device Spartan 6.

REFERENCES

[1]**Rolfes, C., Poschmann, A., Leander, G., Paar, C**2008: Ultra-Lightweight Implementations for Smart Devices – Security for 1000 Gate Equivalents. LNCS, vol. 5189, pp. 89–103. Springer, Heidelberg.

[2**] A. Bogdanov*et al.*,**2007 “PRESENT—An ultra-lightweight block cipher,”in Cryptographic Hardware and Embedded Systems(Lecture Notes inComputer Science), vol. 4727, pp. 450–466.

[3**] Z. J. Shi and R. B. Lee**,2004 “Bit permutation instructions: Architecture,implementation, and cryptographic properties,” Ph.D. dissertation,Depart. Elect. Eng., Princeton Univ., Princeton, NJ, USA,

[4] **J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw**,2011 “The LEDblock cipher,” in Cryptographic Hardware and Embedded Systems(Lecture Notes in Computer Science), vol. 6917. Berlin, Germany:Springer-Verlag,, pp. 326–341.

[5] **A. Bogdanov*et al.*,**2007 “PRESENT: An ultra-lightweight block cipher,”inCryptographic Hardware and Embedded Systems (Lecture Notes in Computer Science), vol. 4727. Berlin, Germany: Springer-Verlag,pp. 450–466.

[6] **Z. Shi and R. B. Lee**,2000 “Bit permutation instructions for accelerating software cryptography,” in Proc. IEEE Int. Conf. Appl. Specific Syst.,Archit. Process. (ASAP), Jul. pp. 138–148.