International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

Implementation of Present Cipher on FPGA for IoT Applications

Anitha Kumari S
M.Tech Scholar,ECE
Dr. Ambedkar Institute of Technology
Bengaluru, Karnataka, India

Dr. Mahalinga V Mandi
Professor Dept.of ECE
Dr. Ambedkar Institute of Technology
Bengaluru, Karnataka, India

***Abstract*— In Internet of Things (IoT),lot of informations is being shared between the devices like smart houses,self driving cars etc. But the information is not protected as the Hackers can hack our system and also steal or change our information.In such circumstance there is a need for protecting the data using cryptographic algorithms.In this Paper a strong Encryption Cipher Called PRESENT which is an Ultra-Lightweight Symmetric Block Cipher is discussed as an application of PRESENT cipher,Text message encryption is considered.**

***Keywords*— *IoT, Lightweight Cryptography, PRESENT Cipher, FPGA*.**

I. INTRODUCTION

The need for Lightweight Cryptography is increasing now-a-days due to its flexible usage in constrained environment [1]. IoT mainly is a communication environment for RFID Tag's and Sensor Nodes technology which are being used in home automation, Healthcare monitoring, structural monitoring etc. The incitation of Lightweight cryptography is to use less memory,less resources and less power supply to contribute for security as well as privacy solution [2].People in the present are surrounded by an environment which is completely dependent on smart devices.Smart devices are mainly dependent on RFID tag's,Sensor nodes,actuators and IoT. As the number of smart devices increases the need for security also increases since it is constrained environment.In such condition a high security cryptographic primitive is needed ,One such example is 'Lightweight Cryptography'.It is tailored mainly for the use in constrained environment.An high security cryptographic algorithm is needed for encryption and decryption of the information in Lightweight Cryptography,One such is Lightweight Symmetric Block Cipher called 'PRESENT CIPHER'[1],[2].In this paper we will discuss the structure, working and implementation of PRESENT.

The rest of the paper is organised as follows

Section II discusses about PRESENT cipher, its Algorithm and Methodology..In Section III the Key Scheduling Algorithm of PRESENT is discussed.Section IV discusses the application of PRESENT Cipher for Text message encryption .Section V reviews the obtained Results.Finally In Section VI the paper is concluded.

II. PRESENT CIPHER ALGORITHM AND METHODOLOGY

A. PRESENT Cipher

PRESENT is an Ultra-Lightweight Symmetric Block Cipher.It was regulated in 2012 by The International Organization for Standardization/The international electro technical commission (ISO/IEC) suitable mainly for Lightweight Cryptography and it is also developed for implementation in constrained environment [1],[3],[4].The main objective of PRESENT Cipher is its extensive use in Lightweight environment such as RFID Tag's,Sensor networks and IoT. It has gained popularity due to its Lightweight nature by providing high level of security in a very small environment with less area,less power,less resources and less memory.Hardware implementation Of Present Cipher is done using FPGA since it is cheaper,easier to use.,uses less power and its reconfigurability nature.

B. PRESENT Algorithm and Methodology

PRESENT is a 31 round Substitution-Permutation network (SPN) based block cipher with block size of 64-bit and 80 bit or 128 bit of key size [1],[2],[5].The 64-Bit keys are generated by internally processing the given 80-Bit input key at each round by the Key Scheduling Algorithm[5] .The input plain text is modified to obtain the encrypted Cipher text by application of confusion and diffusion over the data using three main operations at each round.They are

1. AddRoundKey
2. S-Box layer or Substitution Layer
3. P-Layer or Permutation Layer

The round function and the basic structure of a PRESENT cipher with 64 bit block size and 80 bit Key length is as shown in figure 1

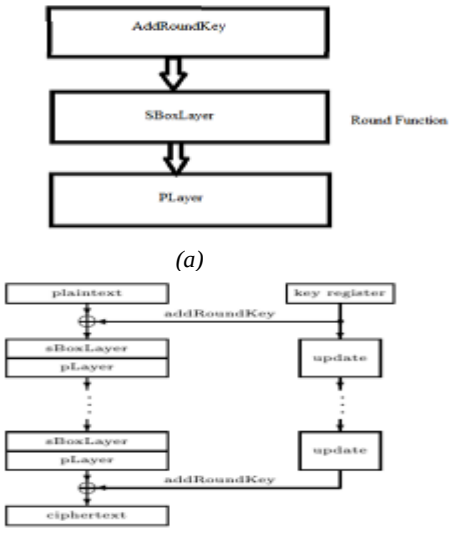

Fig. 1. (a) Round Function in PRESENT (b) Structure of PRESENT

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

The three operations are performed as follows:

1. **AddRoundKey**:It performs the bitwise EX-OR operation between the block of plaintext and the roundkey [1].
2. **Substitution Layer**:It consists of substitution Box (S-Box) which is 4 bit to 4 bit [1] used for the input bit substitution [2].Here the length of the output bits and input bits has to be same.The S-Box table used for substitution is shown in table 1

TABLE I. THE HEXADECIMAL NOTATION OF PRESENT S-BOX

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

3. **Permutation Layer**:Bit permutations takes place in this layer or it is also just a mixing layer (rewiring) [6].Its operation is given by the following table 2 in which the i^{th} bit of the state is moved to P(i) bit positions

TABLE II. THE PERMUTATION OF BITS USED IN PRESENT

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

Previously we had discussed that PPESENT is a SPN based network.The SPN network is used to perform several mathematical operations in Block cipher algorithms.It applies several rounds of S-Box and P-Layer on the block of Plaintext and key given as input alternatively to produce cipher block.The structure of SPN is as shown in the figure 2.

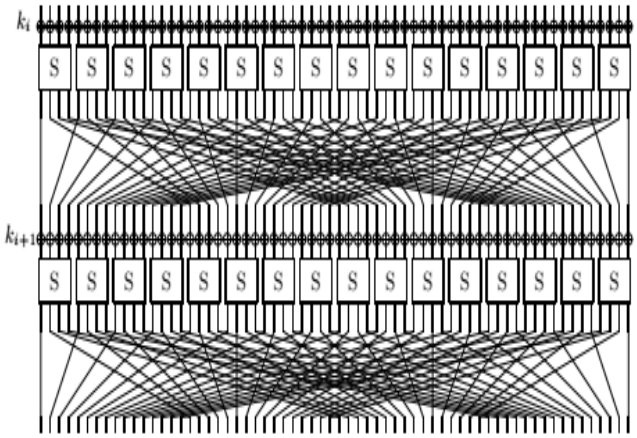

Fig. 2. SPN network

III.KEY SCHEDULING ALGORITHM IN PRESENT

PRESENT supports 80 bit or 128 bit key.In this project we concentrate on 80 bit key.In the Key Scheduling Algorithm initially the user-provided key is stored in the key register given as K.At round i the 64 bit round key is updated with the 64 Most significant bits of the current contents in the Key Register K [2],[7].It is given by

$$Ki=k63\ k62\ldots\ldots\ldots k0 \quad (1)$$

Later the updation of Key register is as follows

1. Rotating the bit positions of Key register by 61 times
2. K18 k17 k16 k15 bits of the key register K are Exclusively-OR'ed with the value of the round counter with the round-counter's LSB on the right
3. The leftmost four bits are passed through the S-Box of PRESENT

The Key scheduling and datapath block of the PRESENT cipher are shown in the below diagram

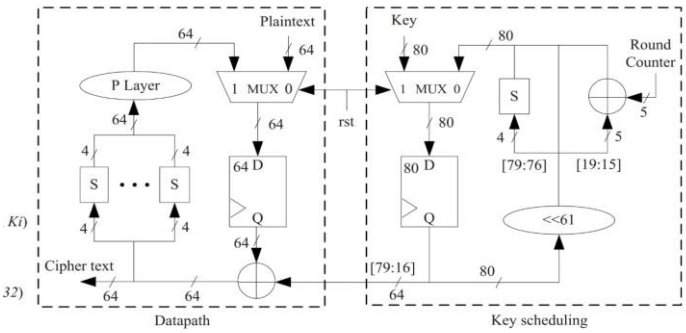

Fig. 3. Datapath and Key Scheduling in PRESENT

IV.APPLICATION TO TEXT MESSAGE ENCRYPTION

In the proposed Application text encryption and decryption is performed by taking input text message and EX-OR'ed with the cipher text key output of PRESENT cipher algorithm to obtain the encrypted text.This part performs encryption of the given text message.In the similar way decryption is also performed by EX-OR'ing the obtained encrypted text with the cipher text key output of the PRESENT cipher algorithm and finally the decrypted plain text is obtained.The Block diagram of the proposed system for text encryption and decryption is as shown in figure 4

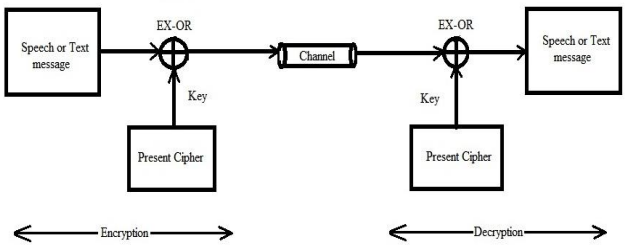

Fig 4. Encryption and Decryption of text message using PRESENT Algorithm

V. RESULTS AND DISCUSSIONS

A.Simulation

The Simulation of the above proposed system was performed in Modelsim software using verilog coding language.The desired encrypted and decrypted text message

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

using PRESENT algorithm was obtained and it is shown in the below figure

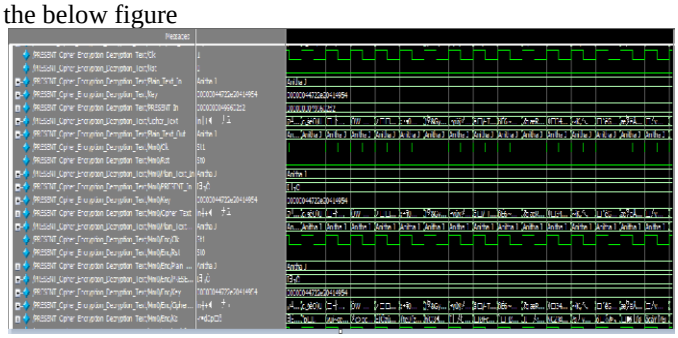

Fig 5. Simulation result of PRESENT cipher for encryption and decryption of text message

A plaintext of ASCII is considered and a 80 bit Hexadecimal key is considered.The encrypted message that is the Cipher text obtained is shown in figure 5.The same 80 bit hexadecimal key is used at decryption to get the Plain text back.

B. *Synthesis*

The synthesis of the proposed application was performed in Xilinx ISE 14.7 using Verilog coding language and the area,delay reports were obtained as shown in figure 6

```
Device utilization summary:
---------------------------

Selected Device : 3s4000lfg900-4

 Number of Slices:                      393  out of  27648     1%
 Number of Slice Flip Flops:            293  out of  55296     0%
 Number of 4 input LUTs:                617  out of  55296     1%
 Number of IOs:                         338
 Number of bonded IOBs:                 338  out of    633    53%
 Number of GCLKs:                         1  out of      8    12%

-----------------------------
```

(a)

```
================================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'Clk'
  Total number of paths / destination ports: 384 / 128
--------------------------------------------------------------------------------
Offset:              10.744ns (Levels of Logic = 3)
  Source:            Enc/M0/Key_Reg_18 (FF)
  Destination:       Plain_Text_Out<2> (PAD)
  Source Clock:      Clk rising

  Data Path: Enc/M0/Key_Reg_18 to Plain_Text_Out<2>
                               Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               7   0.720    1.261  Enc/M0/Key_Reg_18 (Enc/M0/Key_Reg_18)
     LUT3:I1->O            2   0.551    1.216  Enc/Mxor_Cipher_Text_Result<2>1 (Cipher_Text_2_OB
     LUT3:I0->O            1   0.551    0.801  Dec/Mxor_Plain_Text_Out_Result<2>1 (Plain_Text_Ou
     OBUF:I->O                 5.644           Plain_Text_Out_2_OBUF (Plain_Text_Out<2>)
    ----------------------------------------
    Total                     10.744ns (7.466ns logic, 3.278ns route)
                                       (69.5% logic, 30.5% route)

================================================================================
```

(b)

Fig 6. (a)Area (b)Delay report of PRESENT

From the above obtained results it is clear that PRESENT utilizes less resource and delay is also minimum.

Table 3 shows the Area,Frequency and path delay comparisons for different target devices

TABLE III. COMPARISON OF AREA,FREQUENCY AND PATH DELAY OF DIFFERENT TARGET DEVICES

Target device	Area (Slices)	Frequency (MHz)	Path delay (ns)
Xc3s4000l-4-fg900	393	93.07	10.724
Xc6slr45t-4-tqg144	149	210.74	5.194
Xc6slx45t-4-fqg484	149	221.38	4.983

C. *Implementation of PRESENT on FPGA*

Present Cipher was implemented on Spartan 3 FPGA XC3S200-4TQG144C target device.The desired values for encryption and decryption of plaintext were obtained.The obtained value is shown in figure 7

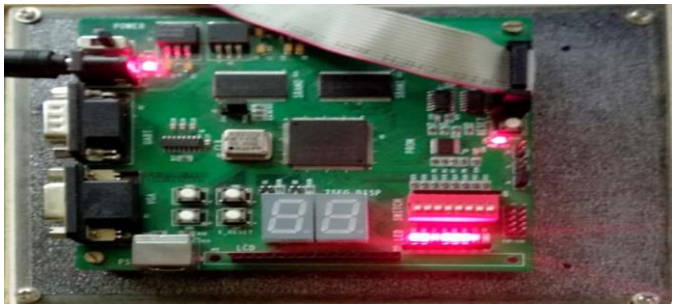

Fig 7. PRESENT implemented on FPGA for encryption and Decryption

VI. CONCLUSION

In the nearing future the need for Lightweight cryptography is very much necessary as billions of smart devices working on IoT environment are emerging.There has to be very high security in such constrained environment.Keeping this uppermost in mind a well developed cryptographic algorithm PRESENT is proposed.It is mainly tailored for lightweight cryptography in constrained environment.Present cipher is also very much advantageous compared to other cipher due to its less resource utilization,Performance and high security nature.In this paper the PRESENT Cipher algorithm was coded,Simulated,synthesised and implemented on FPGA successfully for encryption and decryption of text message..Different target devices were considered for comparison of area,frequency and path delay as shown in table 3.Among the three target devices Xc6slx45t-4-fqg484 utilizes less area of 149 slices,has a path delay of 4.983ns yielding frequency of operation of 200.6 MHz.

VII. REFERENCE

[1] Carlos Andres Lara-Nina;Arturo Diaz-Perez and Miguel Morales-Sandovas.''Lightweight hardware architecture of PRESENT cipher in FPGA'',in IEEE transactions on circuits and System I,2017,(pp 1-7).

[2] X. Guo, Z. Chen, and P. Schaumont.(2008) "Energy and performance evaluation of an FPGA-based SoC platform with AES and PRESENT coprocessors," in Embedded Computer Systems: Architectures, Modeling, and Simulation. Berlin, Germany: Springer,(pp. 106–115).

[3] N. Hanley and M. O'Neill.(2012) "Hardware comparison of the ISO/IEC 29192-2 block ciphers," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI), (pp. 57–62)

[4] A.Bogdanov et al.,(2002).''PRESENT: An ultra- lightweight block cipher," in Cryptographic Hardware and Embedded Systems (Lecture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 08, August-2019

Notes in Computer Science), Berlin, Germany: Springer, (pp. 450–466).

[5] C. A. Lara-Nino; M. Morales-Sandoval, and A. Diaz-Perez .(2016) "Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher," in Proc. Euromicro Conf. Digit. Syst. Design, (pp. 646–650).

[6] T. Xu, J. B. Wendt, and M. Potkonjak.(2014) "Security of IoT systems: Design challenges and opportunities," in Proc. IEEE/ACM Int. Conf. Comput.- Aided Design (ICCAD), Piscataway, NJ, USA, (pp. 417–423).

[7] T. Macaulay .(2017) "Introduction—The Internet of Things," in RIoT Control: Understanding and Managing Risks and the Internet of Things. Boston, MA, USA: Morgan Kaufmann, (pp. 1–26).

[8] D. Dinu; Y. Le Corre; D. Khovratovich; L. Perrin, J. Großschädl and A. Biryukov.(2015) "Triathlon of lightweight block ciphers for the Internet of Things," in Proc. NIST Lightweight Cryptogr. Workshop.

[9] M. Kneževi´c, V. Nikov, and P. Rombouts. (2012) "Low-latency encryption— Is 'lightweight = light + wait'?"' in Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer, (pp. 426–446).

[10] E.B. Kavun and T. Yalcin.(2011) "RAM-based ultra-lightweight FPGA implementation of PRESENT," in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig), (pp. 280–285).

[11] Panasayya Yalla, Jens-Peter Kaps (2009), "lightweight cryptography for FPGAs" in International Conference on Reconfigurable computing and FPGAs.

[12] Hinpreet Kaur and Sakyhivel R,''VLSI Implementation of Lightweight Cryptography Algorithm'',Advances in System Science and Application,vol.16, No.1,pp (95-101), (2016).