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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Design and Implementation of New Lightweight Encryption Technique

M.Sangeetha[1], Dr.M.Jagadeeswari[2]

Student, Department of ECE (PG), Sri Ramakrishna Engineering College, Coimbatore, India[1]

Head of the Department, Department of ECE (PG), Sri Ramakrishna Engineering College, Coimbatore, India[2]

ABSTRACT: Lightweight cryptography is an interesting field which provides security, higher throughput, low power consumption and compactness. Several algorithms like PRESENT, CLEFIA, SEA, TEA, Hummingbird and KANTAN have made the mark to be used as lightweight algorithms. In this paper, we present the design of hybrid lightweight encryption system based on bit permutation instruction GRP (group operation) and the S-box of PRESENT. A new hybrid system is proposed that provides more compact results in terms of both memory space and gate equivalents. This design has resulted in most compact implementation in terms of memory requirement. By using the S-box of PRESENT, it helps in resistant to several differential attacks and also GRP properties are very helpful to attain the compact implementation in hardware. The design is coded using VHDL language and implemented into the Spartan 6 FPGAs.

KEYWORDS: Lightweight cryptography, PRESENT, GRP, Encryption, Bit permutation.

I.INTRODUCTION

THE UPCOMING ERA of pervasive computing will be characterized by many smart devices because of the tight cost constraints inherent in mass deployments have very limited resources in terms of memory, computing power, and battery supply. Here, it is necessary to interpret Moore's law differently. Rather than a doubling of performance, we see a halving of the price for constant computing power every 18 months. Because many foreseen applications have extremely tight cost constraint. For example, RFID in tetra packs over time, Moore's law will increasingly enable such applications. Many applications will process sensitive health-monitoring or biometric data, so the demand for cryptographic components that can be efficiently implemented is strong and growing. For such implementations, as well as for ciphers that are particularly suited for this purpose, we use the generic term lightweight cryptography. Every designer of lightweight cryptography must cope with the trade-offs between security, cost, and performance. It's generally easy to optimize any two of the three design goals security and cost, security and performance, or cost and performance; however, it is very difficult to optimize all three design goals at once. A literature survey shows that, An ISO/IEC standard on lightweight cryptography requires that the design be made with 1000-2000 gate equivalents (GEs) [1].RFID tags may have 1000-10000 GEs out of that only 300-2100 GEs would be available for security aspects [2].Ciphers like AES [3], DES [4][5] would result in high GEs that make them infeasible for small scale real time applications. Light variants of DES such as DESL [6], DESXL [7] have been proposed by slightly modifying the algorithms, by reducing the S-boxes and by using key whitening to increase security levels. Bit permutations are popularly known to be used in permutation block known as diffusion property. We have carefully studied the properties and security aspects of bit permutation instructions like GRP [10].PRESENT is a Substitution Permutation network based on 80 bit or 128 bit key size and 64 bit block size. Alternatively a new structure called PRESENT [8] is a block cipher with 31 rounds and its various variants need 2520 to 3010 GEs to provide adequate security levels.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

II. GRP: A BIT PERMUTATION INSTRUCTION

GRP [10] is one of the most complicated bit permutation instructions that make it an obvious choice to be used in cryptographic environment. GRP performs n bit permutation with $\log_2$ (n) steps while other instructions take O (n) steps [10].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 we put this bit in R1 into the second group. During this process we do not change the relative positions of bits in the same group. Finally, putting the first group to the left of the second group, we get the result value in R3. From the position of two groups, we call the first the left group, and the second the right group.Figure 1 shows how the GRP instruction works on 8-bit systems

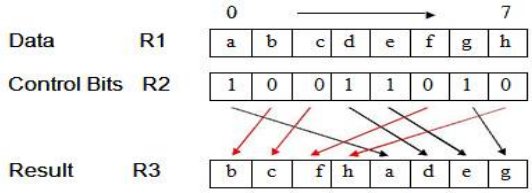

Fig. 1 GRP instruction on 8-bit systems

Papers [10] shows the generation of control bits for the GRP instructions in the integer sequence. In this paper, we have achieved implementation of 128 bit key generation and encryption. GRP algorithm can generate the different keys at different rounds from a given integer sequence. Key generation for corresponding integer sequence is explained with an example in paper [10]. Figure 2 shows the block diagram for 128bit encryption by using GRP.

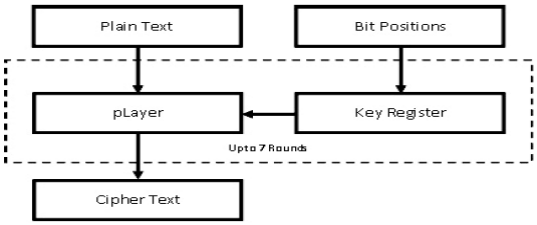

Fig. 2 Block Diagram for 128bit with key generation

III. PRESENT ALGORITHM

In the cryptographic environment, there are two types of instructions, one is the “SP-network” (Substitution Permutation network) like AES, PRESENT etc and other is the “Feistel network” like TEA, XTEA, etc. In this paper, we have focused on SP-network only, as they provide good resistance against most of the attacks. Stream ciphers are also widely studied in the cryptographic environment because of its faster execution, but they are susceptible to attacks compared to SP network block ciphers.

PRESENT structure is Substitution and Permutation (SP) design which consists of S and P layer and consists of 31 rounds. . For this, we referred papers [8], S-box of PRESENT is resistant to brute force attack and has shown good resistance against linear and differential cryptanalysis

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

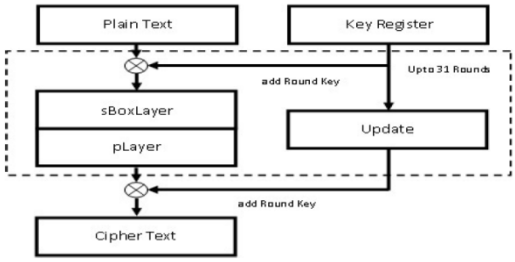

Fig. 4 Block Diagram of PRESENT

Each of the 31 rounds consists of a xor operation to introduce a round key K_i for $1 \leq i \leq 32$, where K_{32} is used for post-whitening, a linear bitwise permutation and a non-linear substitution layer. The non-linear layer uses a single 4-bit S-box S which is applied 16 times in parallel in each round. The cipher is described in pseudo-code each stage is now specified in turn. The design rationale is given in Section 4 and throughout we number bits from zero with bit zero on the right of a block or word.

A. ADDROUNDKEY:

Given round key $K_i = \kappa^i_{63} \ldots \kappa^i_0$ for $1 \leq i \leq 32$ and current state $b_{63} \ldots b_0$, addRoundKey consists of the operation for $0 \leq j \leq 63$,

$$b_j \rightarrow b_j \oplus \kappa^i_j .$$

B. PSEUDO CODE FOR PRESENT:

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

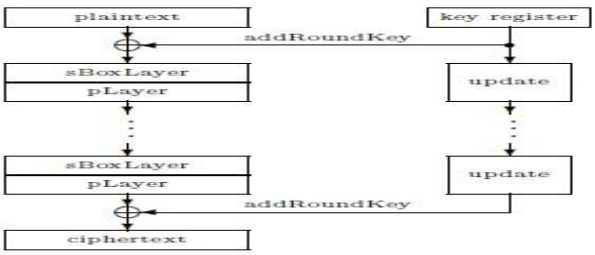

Fig. 5 A top-level algorithmic description of present

C. SBOXLAYER:

The S-box used in present is a 4-bit to 4-bit S-box S: $F^4_2 \rightarrow F^4_2$. The action of this box in hexadecimal notation is given by the following table.

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 1: LOOK UP TABLE OF S-LAYER

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

For sBoxLayer the current state b_{63} . . . b_0 is considered as sixteen 4-bit words w_{15} . . . w_0 where $w_i = b_{4*i+3}||b_{4*i+2}||b_{4*i+1}||b_{4*i}$ for $0 \leq i \leq 15$ and the output nibble $S[w_i]$ provides the updated state values in the obvious way.

D. PLAYER:

The bit permutation used in present is given by the following table. Bit i of state is moved to bit position P (i).

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

Table 2: LOOK UP TABLE OF P-LAYER

In key schedule, Present can take keys of either 80 or 128 bits. However we focus on the version with 80-bit keys. The user-supplied key is stored in a key register K and represented as k_{79} k_{78} . . . k_0. At round i the 64-bit round key $K_i = \kappa_{63}\kappa_{62}$ . . . κ_0 consists of the 64 leftmost bits of the current contents of register K. Thus at round i we have that:

$$\mathbf{K_i = \kappa_{63}\kappa_{62} \ldots \kappa_0 = k_{79}k_{78} \ldots k_{16}}.$$

After extracting the round key K_i, the key register $K = k_{79}$ k_{78} . . . k_0 is updated as follows.

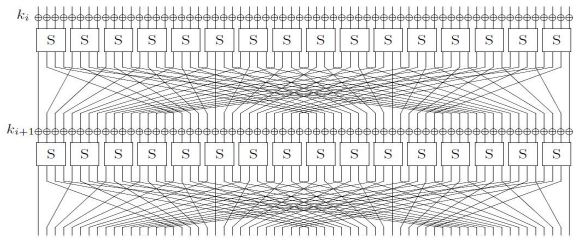

Fig. 6. The S/P network for PRESENT

Thus, the key register is rotated by 61 bit positions to the left, the left-most four bits are passed through the present S-box, and the round counter value i is exclusive-ored with bits $k_{19}k_{18}k_{17}k_{16}k_{15}$ of K with the least significant bit of round counter on the right.

IV.NEW HYBRID MODIFIED GRP ALGORITHM

MODIFIED GRP ALGORITHM

The below figure 7 shows the block diagram of modified GRP algorithm. In which the input is a plain text and it consists of the Permutation layer of GRP and Substitution layer of PRESENT. The algorithm follows the key generation method based on the GRP algorithm. The control bits obtained in the algorithm is acts as a key and accordingly the slayer follows the next step based on the LOOKUP table. Finally the corresponding encryption is performed and cipher text is obtained.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

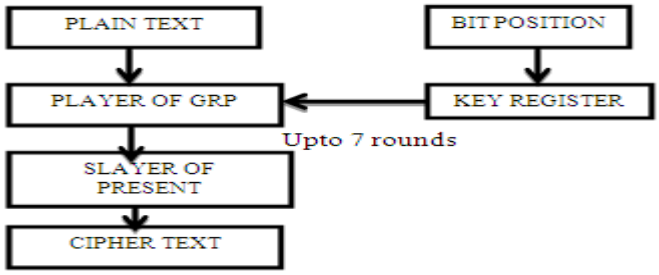

Fig. 7 Block Diagram of MODIFIED GRP ALGORITHM

In PRESENT-GRP module, 64 bit / 128 bit blocks were passed through the S-box of PRESENT and after mapping according to PRESENT, the output was passed to the permutation layer which performed encryption based on GRP algorithms. Keys at each stage were applied based on key generation method of GRP. In GRP key generation, inputs will be the bit positions given by user, and based on that GRP generates a sequence of 0's and 1's which serve as key to the encryption and decryption process. GRP has very robust mechanism of key generation which is the necessity in cryptographic environment. GRP does the key generation as well as encryption with fast bit permutations.

VI RESULTS and DISCUSSIONS

The simulation results of GRP, PRESENT and a modified GRP are designed and compared using Xilinx ISE 14.2 tool and its simulated results are shown below.

A. GRP simulation

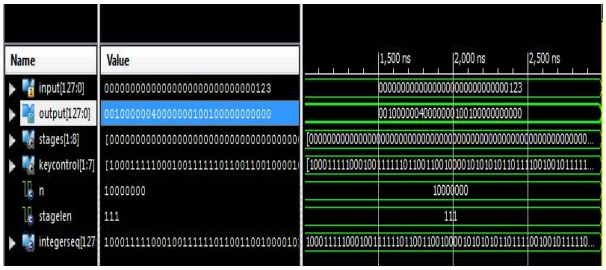

Fig 8: Simulation of **GRP Encryption**. In this the input text of 128 bits is given as 123 and hence the corresponding cipher text is obtained as output.

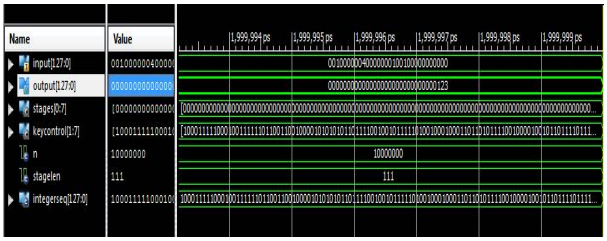

Fig 9: Simulation of **GRP Decryption**. In this the input text of 128 bits is given as the cipher text obtained during the encryption and hence the corresponding plain text 123 is obtained as output.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

B. PRESENT SIMULATION

.

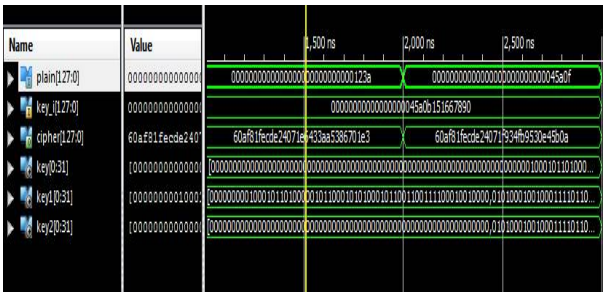

Fig 10: Simulation of PRESENT Encryption. In this, the input text of 128 bits is given as 123 and hence the corresponding cipher text is obtained as output.

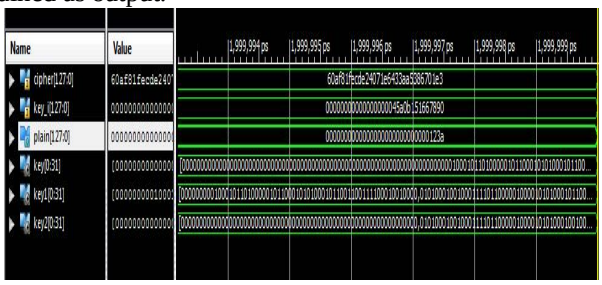

Fig 11: Simulation of GRP Decryption. In this, the input text of 128 bits is given as the cipher text obtained during the encryption and hence the corresponding plain text 123 is obtained as output.

C. MODIFIED GRP ALGORITHM SIMULATION

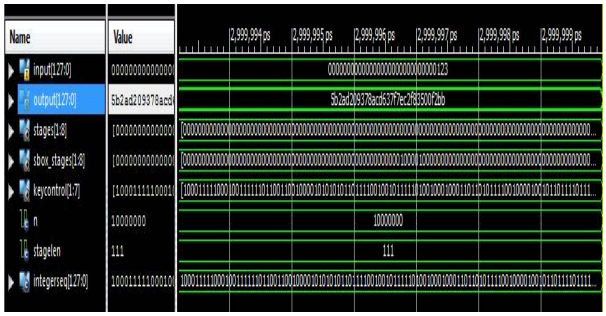

Fig 12: Simulation of Hybrid Modified GRP-PRESENT Encryption. In this, the input text of 128 bits is given as 123 and hence the corresponding cipher text is obtained as output.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

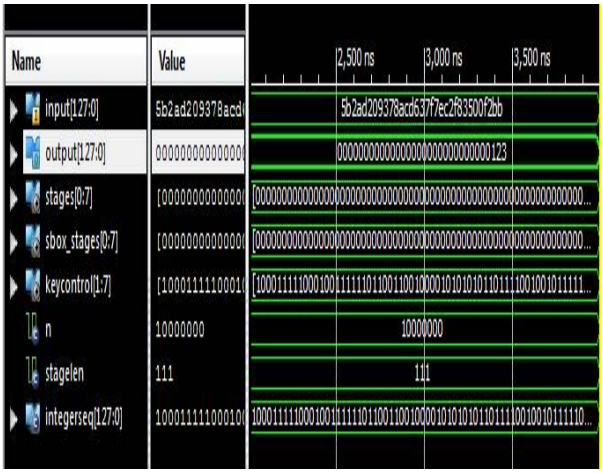

Fig 13: Simulation of Hybrid Modified GRP-PRESENT Decryption. In this, the input text of 128 bits is given as the cipher text obtained during the encryption and hence the corresponding plain text 123 is obtained as output.

Table I

PERFORMANCE ANALYSIS OF ENCRYPTION ALGORITHMS

PARAMETERS	***GPR***	***PRESENT***	***MODIFIED-GRP***
AREA			
NO OF SLICE LUTs	4,080 out of 207360	19840 out of 207360	**3584 out of 207360**
NO OF OCCUPIED SLICE	1,247 out of 51,840	6312 out of 51,840	**896 out of 51,840**
NO OF IOs	384 out of 960	384 out of 960	**256 out of 960**
POWER(W)	3.521	3.526	**3.516**
DELAY(ns)	3.124	64.301	**13.412**

From the above table the three encryption algorithms are compared in terms of area, power and delay and it is noticed that the Modified GRP algorithm has the lesser utilization of device in terms of LUT's slices and power. Hence it is concluded that compared to other algorithms GRP has the minimum number of blocks necessary for achieving the necessary applications. The device Spartan 6 FPGA utilization memory for all the encryption algorithms are shown in table which gives the total resources utilized by each algorithm.

VII. CONCLUSION & FUTURE WORK

The three encryption algorithms GRP, PRESENT and MODIFIED GRP are analysed. The main advantages of the implementations are the minimum numbers of digital blocks used for performing the encryption process in cryptography. Each algorithm performs the operation for 128bit plain text with to the 128bit key and corresponding output text is obtained. These design entry is done through VHDL coding in Xilinx environment and analysed that the

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

modified GRP gives the better results when compared to other modulations and is implemented using Spartan 6 FPGA. The experiment results showed that the AREA (in terms of number of slice LUTs 3584 out of 207360 and in terms of number of occupied slice 896 out of 51,840) and POWER (3.516W) obtained by modified GPR-PRESENT out performs GRP and PRESENT encryption techniques. It can be further extended to several advanced algorithms which possess for low-cost implementation of modulations in the cryptographic environment for higher data transfer and other security purpose.

REFERENCES

[1] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher," In *Cryptographic Hardware and Embedded Systems CHES2011, LNCS*, Vol. 6917/2011, pages 326-341. Springer, 2011.

[2] G. V. Bansod, "Audio Sub word sorter unit on sorter network for sense transmission," In *Proceedings – 2011 IEEE International Conference onCentral System, Computing and Engineering, ICCSCE 2011 Penang*,pages 127-131, January, 2012.

[3] J. Guo, I. Nikolic, T. Peyrin, and L. Wang, "Cryptanalysis of Zorro," In *IACR Cryptology ePrint Archive*, 2013, 713.

[4] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra- Lightweight Cryptography for Resource-Constrained Devices," In *Proceedings of the 14th International Conference on Financial Cryptography and Data Security-FC, LNCS, 2010*, Vol. 6054/2010,pages 3-18, 2010.

[5] Borghoff, Julia, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, MiroslavKnezevic, Lars R. Knudsen, Gregor Leander et al. "Prince–a low-latency block cipher for pervasive computing applications," In *Advances in Cryptology–ASIACRYPT 2012*, pages 208-225. SpringerBerlin Heidelberg, 2012.

[6] X. Yang, M. Vachharajani, and R. B. Lee, "Fast subword permutation instructions based on butterfly networks," In *Proceedings of MediaProcessors 2000 (SPIE 2000)*, pages 80-86, January 2000.

[7]G. Leander, C. Paar, A. Poschmann, and K. Schramm, "A family of lightweight block ciphers based on des suited for RFID applications," In *FSE*, Vol. 4593 of *LNCS*, pages 196-210. Springer, 2007.

[8] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," In *CHES*, Vol. 4727 of *LNCS*, pages 450-466. Springer, 2007.

[9]C. De Canniere, O. Dunkelman, and M. Knezevic, "Katan and ktantan - a family of small and efficient hardware-oriented block ciphers," In *CHES*, Vol. 5747 of *LNCS*, pages 272-288. Springer, 2009.

[10] Z. Shi and R. B. Lee, "Bit permutation instructions for accelerating software cryptography," In *Proceedings of the IEEE InternationalConference on Application Specific Systems, Architectures and Processors (ASAP 2000)*, pages 138-148, July 2000.

[11] A. Poschmann. "Lightweight cryptography: cryptographic engineering for a pervasive world," In Ph. D. Thesis. 2009.