International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June-2017 www.irjet.net p-ISSN: 2395-0072

DESIGN OF A NEW LIGHTWEIGHT ENCRYPTION FOR EMBEDDED SECURITY

Anitha Sekhar[1], Premanand B [2]

[1]*Mtech scholar, Department of ECE, GECI, Kerala, India*
[2]*Assistant Professor, Department of ECE, GECI, Kerala, India*

Abstract - *The incrementing utilization of extensive devices in the field of electronics has raised the concerns about security. In embedded applications, implementing a developed cryptographic environment would not be practical because of the constraints like power dissipation, area and cost. Due to these constraints, the focus is on utilizing lightweight cryptography that requires as less recollection space as possible. Lightweight cryptography is an interesting field that provides a perfect balance in security, low-power consumption, and compactness. In recent years, many compact algorithms like CLEFIA, SEA, TEA, LED, MCRYPTON and PRESENT are used as lightweight crypto engines. .Using the S (substitution) box of PRESENT algorithm, replacing the random permutation in P layer by GRP with confusion property, because all the existing algorithms using bit permutation instructions do not have this confusion property. A new hybrid system called PRESENT GRP is proposed that is more compact in terms of both gate equivalents and memory space. To secure password, Token is used as an additional input to the key, in order to avoid precomputational attacks such as rainbow attack. The hardware simulation is done in Xilinx Spartan 3E FPGA using Xilinx 14.2 Vivado Design Suite.*

Key Words: *Lightweight cryptography, PRESENT, GRP, Encryption, PRESENT GRP.*

1. INTRODUCTION

On a new computing environment called "Internet of things (IoT) "or "Smart objects" networks, a lot of constrained devices are connected to the internet. These constrained devices interact with each other through the network, so the security of the constrained end nodes is paramount. Cryptography is the study of techniques cognate to aspects of information security. Hence cryptography is concerned with the ciphering (encoding) and deciphering (decoding) of information in secret code.

A cipher is a secret method of writing, as by code. Variants of cryptographic algorithms have been designed to achieve different functions of security such as confidentiality, authentication, and data integrity. Conventional encryption cipher uses a single key for both encryption and decryption. Modern protocols will utilize a private key for encryption and a different public key for decryption. These two keys are mathematically cognate in a fashion that allows them to encrypt/decrypt the same information successfully. The key is a value independent of the plaintext. The algorithm will produce a different cipher depending on the specific key at different rounds. Once the cipher text is produced, it may be transmitted. The cipher text can be retrieved back to the original plaintext by using a decryption algorithm and the same key that was utilized for encryption. The security of conventional encryption depends on two factors:

1. The Encryption Algorithm- It must be powerful enough that it is impractical to decrypt a message on the basis of the "cipher text" alone.

2. Secrecy of the key-The security of encryption of the information mainly depends on the secrecy of the key, not the secrecy of the algorithm.

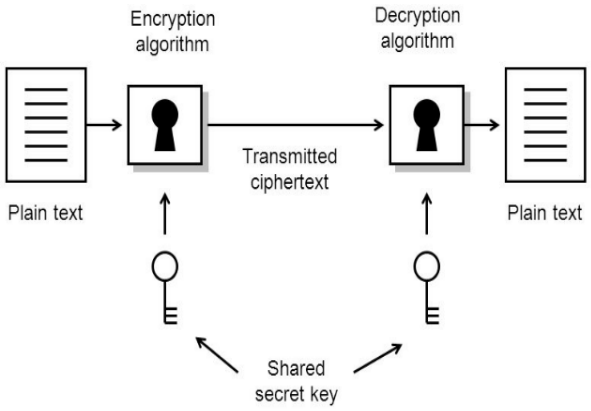

Fig -1: Conventional convolutional encryption

Lightweight cryptography is an interesting field that provides a perfect balance in security, low-power consumption, and compactness. In recent years, many compact algorithms like CLEFIA, SEA, TEA, LED, MCRYPTON and PRESENT are used as lightweight crypto engines [1].

The incrementing utilization of extensive devices in the field of electronics has raised the concerns about security. In embedded applications, implementing a developed

cryptographic environment would not be practical because of the constraints like power dissipation, area and cost. Due to these constraints, the focus is on utilizing lightweight cryptography that requires as less recollection space as possible. The main criterion for the lightweight cipher is to have less memory space and that which would result into a less Gate Equivalent (GEs) count for an efficient hardware implementation without compromising the requisite of vigorous security properties. The proposed circuit introduced in section 3. Section 4 presents the simulation results of the proposed and modified circuit. Finally, concluded in section 5.

2. EXISTING PRESENT ALGORITHM

PRESENT algorithm uses a block size of 64 bits and two key lengths of 80 and 128 bits are supported. Number of rounds used in PRESENT is 31 and is an example of a SP-network. Each of the 31 rounds consists of an 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 [2].

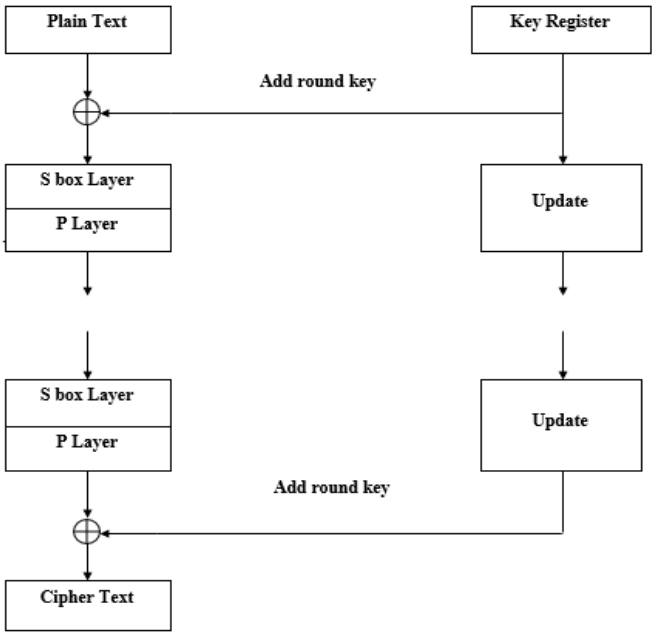

Fig -2: A top-level algorithmic description of PRESENT

The substitution layer uses a single four-bit S-box which is applied 16 times in parallel in each round. PRESENT algorithm is used in situations where low-power consumption and high chip efficiency is desired. The addRoundKey takes a round key and the current cipher state, combines them together using bitwise exclusive-or. The S-box used in PRESENT is a four- to four-bit S-box which is applied 16 times in parallel across the 64-bit block. The S-box of PRESENT is very compact uses 4x4 box in order to minimize the gate complexity and the power consumption. Higher S-box has increased Boolean equations resulting high gate count. Similarly lower bit S-box has less Boolean equations resulting less gate count. The S-box in hexadecimal notation is given by the following table.

Table -1: S Box for PRESENT algorithm [2]

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

Finally the bit permutation used in PRESENT is given by the following table where bit i of STATE is moved to bit position P(i). While it is primarily intended for 80-bit keys, PRESENT can take keys of either 80 or 128 bits.

Table -2: Bit permutation in PRESENT algorithm [2]

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

2.1 KEY SCHEDULE

The user supplied key is stored in a key register K and represented as k_{127}, k_{126} ....k_1, k_0. The key register is rotated by 61 bit positions to the left, leftmost eight bits are passed through two PRESENT S-boxes and Round counter is exclusive-ored with bits k_{66} k_{65} k_{64} k_{63} k_{62}.

For reverse path, leftmost eight bits from the key register are passed through two PRESENT inverse S-boxes, Round counter is counter is exclusive-ored with bits k_{66} k_{65} k_{64} k_{63} k_{62} and remaining bits is rotated by 61 bit positions to the right. Thus the sub keys generated during encryption process is applied as the input to the reverse path and get the output as master key.

3. PROPOSED PRESENT GRP

Using the S-box of PRESENT algorithm, replacing the random permutation in P layer by group operation GRP with confusion property, because all the existing algorithms using bit permutation instructions do not have this confusion property. A new hybrid system called

PRESENT GRP is proposed that is more compact in terms of both gate equivalents and memory space.

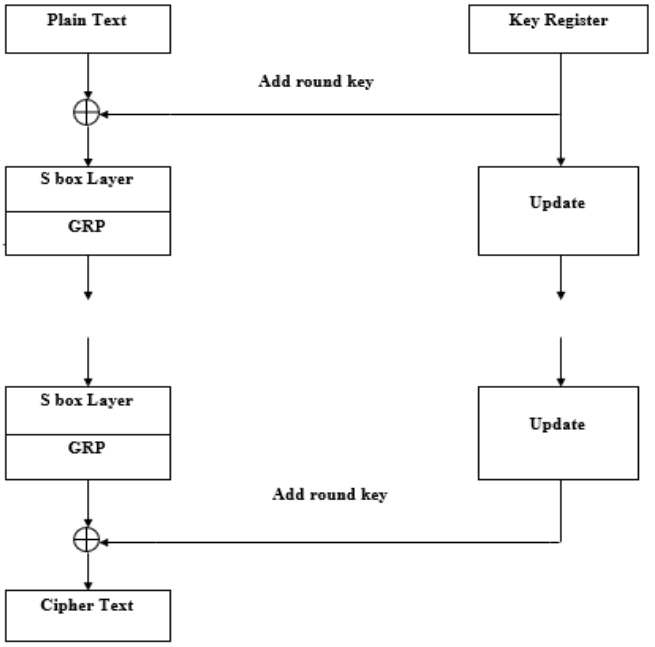

Fig -3: A top-level algorithmic description of PRESENT GRP

The basic concept of the GRP instruction is to divide the bits in the source (data) R1 into two groups according to the control bits in R2.The control bits R2 is used to control each bit in data R1. If the value of bits in R2 is 0, we put the corresponding value of bit from R1 into the first group. Otherwise we put the corresponding bit from R1 into the second group. The result bit is divided into two groups, we call the first the left group, and the second the right group [3]. By doing the concept of GRP, the speed of the system can be improved. Figure 4.2 shows how the GRP instruction works on 8-bit systems.

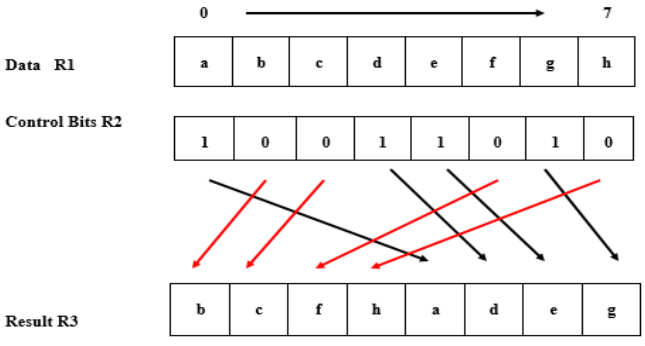

Fig -4: The GRP Instruction on 8 bit systems [3]

GRP is known for fast bit permutation. Simple look up table is an alternative to bit permutation, but permutation operations are far better than a look up table. A Look up table requires nearly 23 numbers of instructions for a 64 bit permutation while GRP require only six lines of instructions, GRP is complex in nature that makes it more suitable for cryptographic environment as compared to operations like addition, shifting or multiply. GRP is suitable in an application like remote sensor continuously encrypting data and sending it to a server location.

3.1 GRP implementing encryption for 8 bit

GRP algorithm can generate the different sequences of instruction at different rounds from a given integer. GRP 128 bit permutation is designed and implemented where the plain text is in the range of 128 bit and GRP performs the exact same operations for 128 bits which is performed for an 8 bit encryption.

Let us assume that the word length of input w = 128 that needs to be permuted with the help of GRP. To permute 128 bits, the operation needs total number of 7 stages as $2^7 = 128$ bits. Similarly, for 8 bit and 64 bit permutations, we need total of 3 and 6 stages, respectively. The output from the first stage of GRP encryption is given as the input to the second stage, similarly output from the second stage of GRP encryption is given as the input to the third stage.

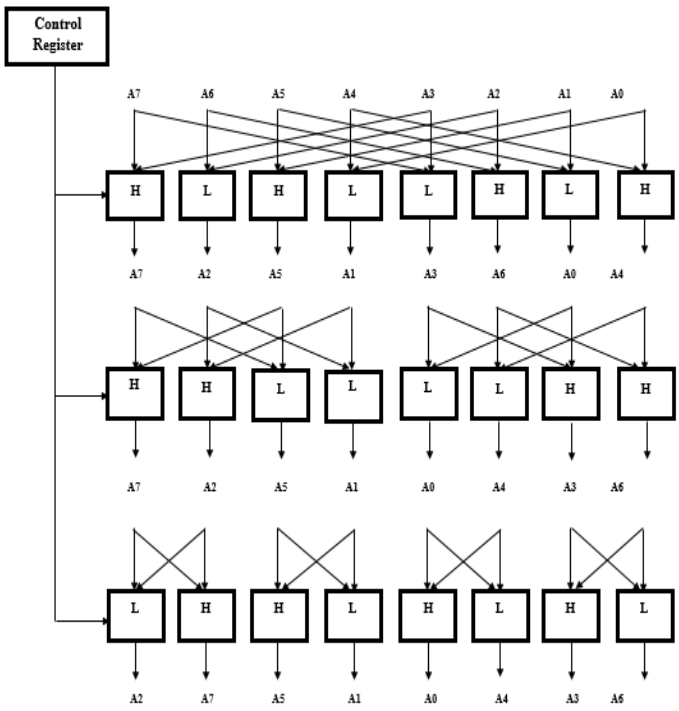

Fig -5: GRP implementing encryption for 8 bit

It can be obtained by using this equation $2^n = w$ where w indicates word length and n indicates number of stages, if word length is 128 bits then P value will be 64 which represent in the first stage of permuting 128 bits, first group is the 0^{th} bit and 64^{th} bit second will be 1^{st} and 65^{th} bit, third group will be 2^{nd} and 66^{th} bit and similarly last will be 63^{rd} and 127^{th} bit. Decryption structure has been

designed for given arrangement and three control words are applied to the structure to get the original information back. The output from the first stage of GRP decryption is given as the input to the second stage, similarly output from the second stage of GRP decryption is given as the input to the third stage.

3.2 MODIFIED PRESENT GRP

Token is a random data this is used as an additional input to the password, in order to avoid precomputational attacks such as rainbow attack. A linear transformation is done for generating token. It is mainly used to stop someone from precomputing a reverse lookup table that allows an attacker to find a password. So the total possible combinations will be more that is in the range of 2^{256} and therefore difficult to crack.

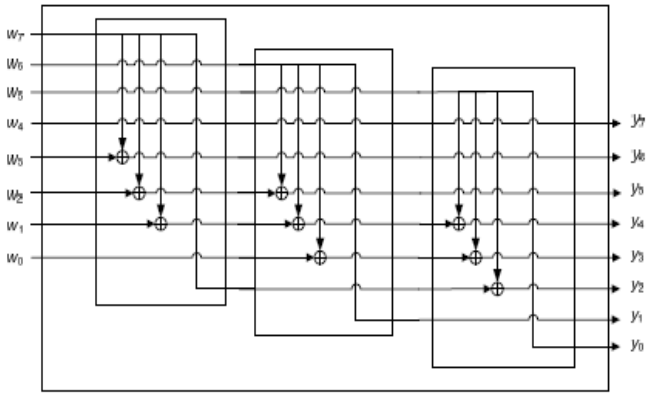

Fig -6: Components of linear transformation

4. SIMULATION RESULTS

The 128 bit of plain and key are the inputs applied to the Proposed PRESENT GRP and after encryption process, generate outputs as 128 bit cipher and sub key. Fig -7 shows the simulation result of Proposed PRESENT GRP encryption on 128 bit.

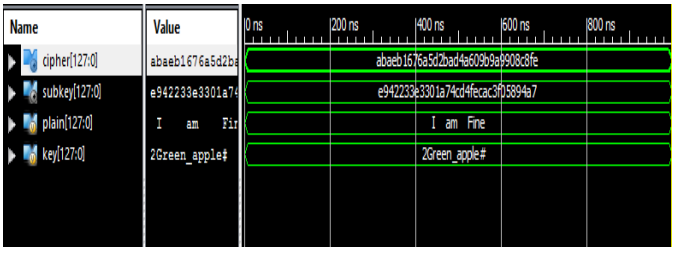

Fig -7: Simulation result of Proposed PRESENT GRP encryption on 128 bit

Fig -8 shows the simulation result of Proposed PRESENT GRP decryption on 128 bit. The 128 bit of cipher and sub key are the inputs applied to the Proposed PRESENT GRP and after decryption process, generate outputs as 128 bit plain and key.

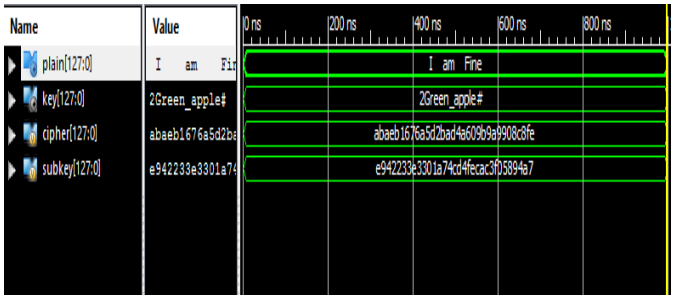

Fig -8: Simulation result of Proposed PRESENT GRP decryption on 128 bit

Fig -9 shows the simulation result of Modified PRESENT GRP encryption on 128 bit. The 128 bit of plain and key are the inputs applied to the Modified PRESENT GRP and after encryption process, generate outputs as 128 bit cipher and sub key.

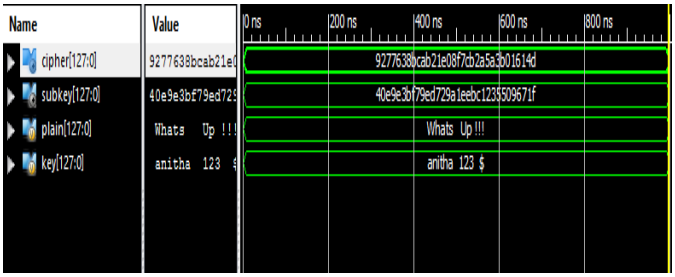

Fig -9: Simulation result of Modified PRESENT GRP encryption on 128 bit

Fig-10 shows the simulation result of Modified PRESENT GRP decryption on 128 bit. The 128 bit of cipher and sub key are the inputs applied to the Modified PRESENT GRP and after decryption process, generate outputs as 128 bit plain and key.

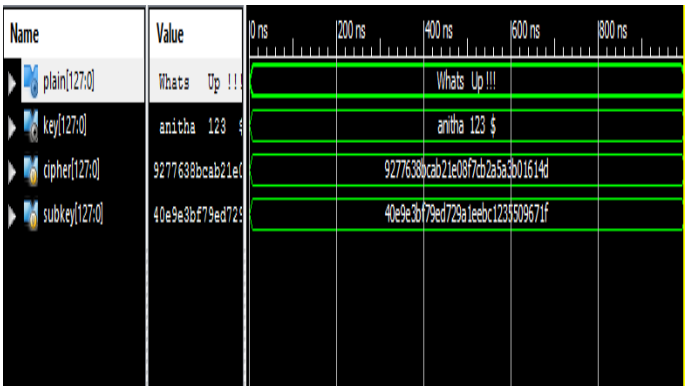

Fig -10: Simulation result of Modified PRESENT GRP decryption on 128 bit

The simulation results were compared in table: 3.

Table-3: Comparison Results of PRESENT, Proposed PRESENT GRP and Modified PRESENT GRP

Algorithm	PRESENT	Proposed PRESENT GRP	Modified PRESENT GRP
Delay(ns)	12.21	10.34	18.19
Power(mW)	320	240	300

Power can be reduced by 25% using PRESENT GRP. System delay is more for Modified PRESENT GRP in the range of 18.19ns, but is shows good resistance to attacks.

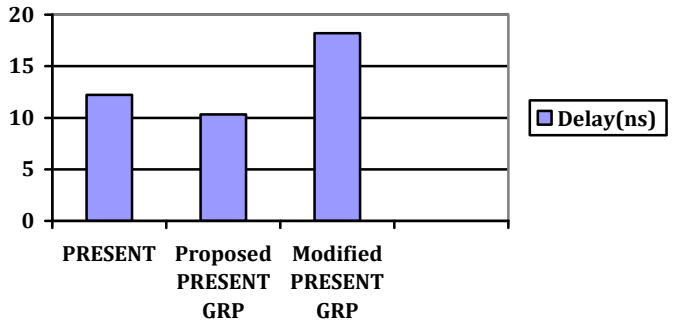

Chart-1: Delay Analysis

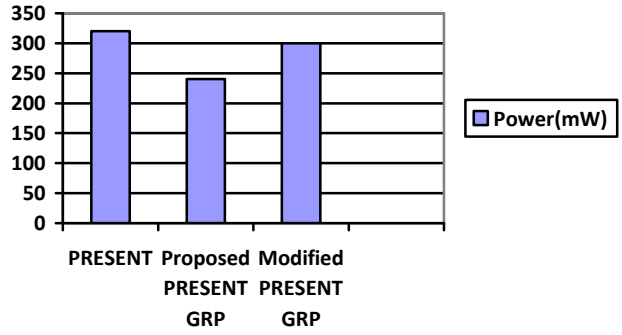

Chart-2: Power Analysis

5. CONCLUSIONS

A new hybrid system called PRESENT GRP is designed that is more compact in terms of both gate equivalents and memory space. Power can be reduced by 25% using PRESENT GRP. To secure password, Token is used as an additional input to the password, in order to avoid precomputational attacks such as rainbow attack. System delay is more for Modified PRESENT GRP, but is shows good resistance to attacks. The hardware simulation is done in Xilinx Spartan 3E FPGA using Xilinx 14.2 Vivado Design Suite.

REFERENCES

[1] Gaurav Bansod, Nishchal Raval, and Narayan Pisharoty, "Implementation of a new lightweight encryption design for embedded security", IEEE Transactions on information forensics and security, Jan 2015 vol. 10, pp. 142-151.

[2] A. Bogdanov et al., "PRESENT- An ultra-lightweight block cipher, in cryptographic hardware and embedded systems", vol. 4727, Springer-verlag,pp. 450-466, 2013.

[3] Li G. Y, Xu Z. K, Xiong C, et al., "Bit permutation instructions: Architecture, implementation, and cryptographic properties", IEEE, pp. 28-35, 2012.

[4] T. Eisenbarth and S. Kumar, "A survey of lightweight cryptography implementations", IEEE Des. Test. Comput., vol.24, no. 6, pp. 522-533, Nov 2012.

[5] G. Bansod. G. Aman, G. Arunika, B. Gajraj and A. Harshita, " Experimental analysis and implementation of bit level permutation instructions for embedded security", WSEAS Trans. Inf. Sci. Appl., vol. 10, no.9, pp.303-312, 2011.

[6] X. Yang and R. B. Lee, "Fast sub word permutation instructions using flip network stages", in Proc. Int. Conf. Comput. Design, pp. 15-22, Sep 2012.

[7] G. Leander, C. Paar and K. Schramm, "A family of lightweight block ciphers based on PRESENT suited for RFID applications", in Proc. FSE, vol. 4593, pp. 196 -210, 2012.

[8] C. E. Shannon, "Communication theory of secrecy systems", Bell Syst. Tech. J., vol. 28, no. 4, pp. 656-715, 2010.

[9] M. Vacharajani and R. B. Lee, "Fast sub word permutation instructions based on butterfly networks", Proc. SPICE, vol. 3970, pp. 80-86, Jan 2011.