International Journal of Pure and Applied Mathematics
Volume 119 No. 14 2018, 1825-1832
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

AN EFFICIENT IMPLEMENTATION OF PRESENT CIPHER MODEL WITH 80 BIT AND 128 BIT KEY OVER FPGA BASED HARDWARE ARCHITECTURE

Hengameh Delfan Azari[1], Dr. Prashant V Joshi[2]
[1]School of ECE, REVA University, Bengaluru, India
[2]Assistant Professor, School of ECE, REVA University, Bengaluru, India

ABSTRACT: The Cryptographic architecture involves the different functional models which help to achieve better security against internet attacks. Nowadays, various cryptographic techniques were presented to protect against network generated or network spread malicious faults. Among the existing techniques it is been quite difficult task to identify an efficient techniques for a particular application. The lime weight standardized cipher models were mainly considered for solving the issues and constraints of the security. This paper presents a PRESENT Cipher model which incorporates both the encryption as well as decryption process by using 80bit and 128bit key for 64bit input data security at hardware level. The design of the model is performed by utilizing Verilog-HDL on Xilinx 14.7 ISE platform and is implemented over Artix-7 FPGA board. The performance analysis is performed by comparing both the key generated results with respect to slices, LUT's, Flip-flops (FF's), Frequency and throughput as performance parameters and is found that throughput is improved with PRESENT cipher model.

Keywords: Cryptography, Hardware Architecture, FPGA, PRESENT cipher, Security, Verlog.

INTRODUCTION

IOT is a technology where large numbers of devices are controlled over the internet through programs or by the consumer directly. As the broadband internet is now usually available and the cost of connectivity also decreased, more sensors are connected to it [1]. The practical comprehension of the IOT scheme depicted to lot of challenges such as power constraints, area constraints, environment constraints, integration of interdisciplinary domains, interfacing requirements of objects. Therefore, among all these significant consideration for the recognition of IOT in cyber-attack world is safety [2]. The possible of cyber-attack or malicious attack are highly spread and sometime it is uncontrollable if object are connected via internet. The malicious of attacks may directly affect the whole physical world. The process of sharing large amount of information continuously with the devices must securely interact with the IOT base form. IOT systems have a continuous procedure of data collection and exchanging data from different servers. It is essential to maintain confidentiality, authentication, and data integrity. Hence, data security is a main thing that wants to give importance in IOT [3].

Generally, IOT system includes hardware, middleware and presentation layers. The hardware layer provides actuators and sensors, middle layer gives computation with storage and final layer, presentation layer comprise interpretation tools for taking information from various base form. The data security in IOT is formulated in physical or semantic or computation layer. Most of the popular novel attacks are executed on physical layer and major attacks are performed on software layer [4]. Ex: Sybil attack on RFID, which comes under physical level. The whole system is secured by security implementing at lowest level that is physical layer because of this the architecture of IOT schemes cannot afford for realizing the security in middle and presentation levels due to computational restraints. The IOT devices are utilizing less hardware and devices are battery powered. Therefore security solution at hardware level requires algorithm with small footprint which all comes under the energy budget [5, 6, 7].

Lightweight cryptography gives a solution to security implementation in hardware level. These cryptographic methods specially established for embedded systems. The use of IOT devices increased all over the world because of lightweight technique used in IOT devices which also gives an end to end communication security under computation, memory limits and low power consumption. So lightweight hardware technique with PRESENT cipher highly applicable for extreme resource constrained applications and when AES algorithm is unsuitable application. They are used with challenge-response authentication protocol, encryption and decryption of communication in counter mode [8, 9].

These security issues can be tackled by using PRESENT Cipher model which is a standardized one. The designed PRESENT cipher model is implemented by using 80-bit and 128-bit key, for 64-bit data input. Finally, this paper is organized with sections like literature survey incorporated with existing researches (Section 2), basics of Block cipher model (Section 3), proposed cryptographic technique (PRESENT Cipher model using 128 and 80bit key) is given in Section 4, analysis of the obtained results with performance analysis in discussed in Section 5, finally the conclusion is given Section 6.

1. LITERATURE SURVEY

This section provides the discussion on various existing researches in cryptographic technologies for security enhancement at hardware level.

The system integration of AES and PRESENT coprocessors was performed in Guo et al. [10] where the system outline analysis was performed by simulating the system model over FPGA based System on Chip (SoC). The outcomes of the system performance, energy and implementation were estimated for both PRESENT and AES which suggests that PRESENT less energy efficient than AES in a lightweight block cipher with lesser security level.

The light weight block cipher PRESENT implementation on FPGAs was presented in research work of Sbeiti et al. [11]. The design of PRESENT cipher algorithm minimum hardware footprint was exhibited. The outcome of the scheme produced more efficiency, so that PRESENT is well suitable for high throughput and high speed presentations.

The work of Yalla and Kaps [12] introduced lightweight cryptography for FPGAs. In this method block cipher independent optimization technique was presented for Xilinx Spartan3 FPGAs which all put on to the PRESENT and HIGHT lightweight cryptographic algorithm. With the analysis of proposed scheme the ratio of throughput and area in PRESENT gives 240kbps/ slice and HIGHT with 720kbps/slice.

The Differential Power Analysis (DPA) attacks methods were investigated and to check the feasibility PRESENT algorithm was introduced in the study of Duan et al. [13]. The analyzed result gives indication that the proposed algorithm is vulnerable to power consumption occurrence under hardware settings i.e., DPA is more efficient against PRESENT algorithm.The work of Guo et al. [14] introduced a Negative Bias Temperature Instability (NBTI) for calculating the effects on power analysis attacks. The implementation of PRESENT algorithm indicates that the base line circuit were not suitable for aged circuit. The analysis of present method concludes that the Classical CPA attacks are not significantly affected by aging while template rate of attack changes significantly.

The S-box architecture for secure data and implementation of cryptographic hardware was presented in work of Rahaman et al. [15]. In this method C-testable S-box was invited for data encryption which is most complex block in hardware implementation so S-box structure was divided into a Read-Muller form and these are tested by using BIST circuit. The proposed architecture was efficiently evaluated against S-box functionality.

The FPGA implementation of the Ultra lightweight cipher PRESENT algorithm was evaluated in research work of Kavun and Yalcin [16]. In the first design S-boxes were utilized within the slices and next design these all combined into the same RAM box used for state storage, which all more suitable for lightweight applications. The outcome of the proposed method produced reasonable throughput is 6.03kbps and 5.13kbps at 100 kHz, low cost and low area.

The study of Tay et al. [17] introduced PRESENT algorithm with Boolean S-box was implemented on to FPGA. The methodology used 8 bit data path to decrease size of hardware. Boolean S-box was used instead of using LUT based S-box for implementing system. The outcome of proposed scheme achieved smallest FPGA implementation of the PRESENT cipher to date and throughput of 51.32Mbps at 236.574 MHz's. A FPGA based hardware structure in terms of area for the PRESENT lightweight block cipher was presented in work of Nino et al. [18]. The experimental results based on area PRESENT method exhibits smaller latency and lesser FPGA resources compare to other existing method.

The research study of Chouhan [19] introduced a PRESENT cryptographical algorithm for message encryption in low cost reconfigurable hardware NETFPGA 1G. NETFPGA contains all the logic resources, memory and Gigabit Ethernet interfaces to build a complete router, switch and other security devices. PRESENT algorithm was used here to enhance the security in NETFPGA. The scheme was analyzed with high efficiency by algorithm implementation. From the above survey analysis most of the researches were considered the conceptual implementation of

different cipher techniques. Also, it is found that very rare works were considered the FPGA implementation to achieve hardware level security.

2. BLOCK CIPHER MODEL

The block cipher PRESENT is a Substitution Permutation Network with block size of 64 bits which operates 31 rounds and final round operates only with two different keys. The PRESENT cipher considered a key with key sizes of 80 bits and 128 bits. PRESENT cipher gives a solution to low security use such as for tag based deployments and other hardware based application.

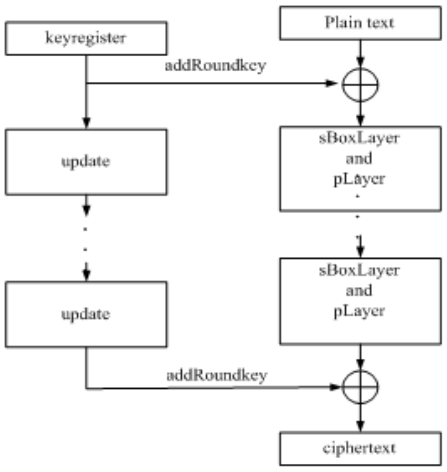

Figure 1 Block diagram of PRESENT ciphering

In the PRESENT ciphering round key (RK) Km, 1≤ Km ≤32 where m= 1, 2…... 31 and, is applying for each of the 31 rounds for functioning bitwise XOR operation. K32 is the additional key which is used for linear bitwise permutation and non-linear substitution. The S-BOX of 4 bits is used in non-linear layer which is parallelly operated 16 times in each round. The above figure.1, gives the description of PRESENT cipher is explained below based on pseudo code.

First, consider the block of the plain text is STATE to explain PRESENT cipher generation.

1. Generate→RK ( )
2. for n=1,2,3, ……. 31

add(RK)→(STATE, K_n)
sbox(STATE)
pbox(STATE)
end for
3. add(RK)→(STATE, K_{32}); where RK is Round Key

A. Add Rk

For the given round key operates 31 rounds i.e. Kn $=K_{63}^n \ldots . K_0^n$, $1 \leq n \leq 32$ and STATUS $a_{63}^n \ldots . a_0^n$ round key added to the substitution permutation layers. The round key bits and plain text STATUS bits are added by XOR operation. Here the operation results of XOR are applied to next iteration to perform SP process. The bitwise XOR operation is defined as

$$a_n = a_n \; XOR \; K_n^m$$

B. S-BOX

The substitution layer consists of 4 bit inputs and 4 bit outputs of each 16 S-BOXes. This process can be represented as mapping operation denoted as S: $F_2^4 \rightarrow F_2^4$. The S-BOX example is illustrated in below table 1 in hexadecimal notation based on given inputs.

Table1: S-BOX process

x	0	1	2	3	4	5	6	7	8	----	F
S[x]	C	5	6	B	9	0	A	D	3	----	1

The current STATE $a_{63}^n \ldots . a_0^n$ is taken as 16 4bit words

$w_{16} \ldots . w_0$

where $w_n = a_{4*n+3} || \; a_{4*n+2} || \; a_{4*n+1} || \; a_{4*n}$ for $0 \leq n \leq 15$.

The output nibble S[w_n] gives the updated values.

C. Permutation- layer

The permutation of bits is done by moving a bit n of STATE to bit position P (n) as shown in the table 2. We can observe here the bit on position 0 of input moved to the P-layer position 0 and bit on position 1 moved to bit position 4 of p-layer so on.

Table 2 The bit permutation using 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

D. Key Schedule

The key length is considered to be 80 bits and user supplied key is saved in a register which is called as key register 'K'. The key register is represented as K79 K78 K77…. K0. At round 'n', the round key of 64 bits Kn=$K_{63}^{n} \ldots K_{0}^{n}$ consists of 64- leftmost bits of K state register contents. The round key at round 'n' is Kn = K79 K78 K77…. K16. The updated key register K is as follows

1. [K79 K78 K77…. K0] = [K18 K17 …. K19]
2. [K79 K78 K76K76] = S[K79 K78 K77 K76]
3. [K19 K18 K17 K16 K15] = [K19 K18 K17K16 K15] XOR round_counter

The key register 'K' is rotated left by 61 bit positions, 4 bits of left most position send to PRESENT S-BOX and 'n' value of RK is XORed with bits K19, K18, …, K15 of K with round counter LSB on the right.

3. PROPOSED CRYPTOGRAPHIC TECHNIQUE

The present module is a kind of Substitution Permutation (SP) network consisting of 31rounds with block length being 64 bits supporting key length 80 bit. The figure.2 illustrated below represents the top module of PRESENT algorithm that comprises of clock, k_load to initiate the process of data encryption, d-load, key-input which is 80bit in size. The size of the input fed is din to be 64 bits.

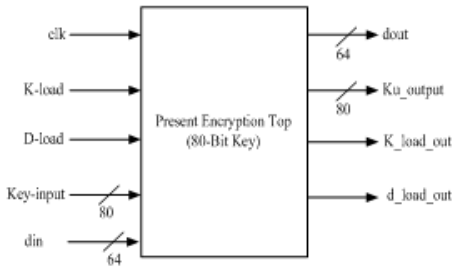

Figure 2 Top module of Present Encryption 80 bit Key block cipher

The hardware architecture of PRESENT encryption is represented in figure.3. It uses 64 bit of inputs, 64 bit of outputs and 80 bit of key i.e. the key size used 80 bit. A single 64-bit register stores the state along with key of 80 bit register which service parallel input and multiple bits shift. There is need of 16 cycles in order to load the data to processing of data of 64 bits plain text block. This procedure gives the outputs.

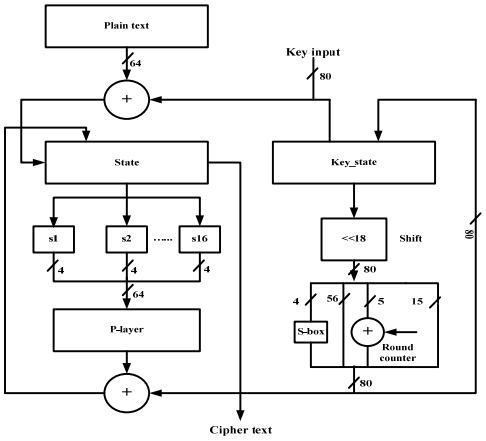

Figure 3 Hardware architecture of Present Encryption 80 bit Key block cipher

The PRESENT decryption block cipher hardware architecture is as shown in below figure 4. The updated value of key is XORed with the 64 bit value of cipher text or encrypted data. The 64 bit XORed output is saved in state register and it starts counting from 31 till the value reaches zero. Inverse p-layer (Permutation layer) performs operation likewise; 64 bit data is splits into 16 inverse S-BOXes each contains 4 bit. End of the first round in the design the count reduces by one i.e. 31 to 30. Likewise the procedure repeats itself until the value reaches 0 and parallelly the updated 80 bit value of key update its count info in the state register. The value is shifted twice, the 80 bit attained data split into4 bits each for one inverse S-box, 5 bits used for round counter, 15bits and 56 bits forwarded for XOR operation with 64 bit inverse permutation layer output. The start register output reaches 0 counts that is plain text.

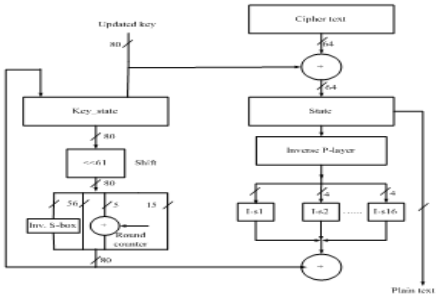

Figure 4 Hardware architecture of Present Decryption 80 bit Key block cipher

Similarly, the top module of 128 bit key based PRESENT cipher model is having clock, k_load to initiate the process of data encryption, d-load, key-input which is 80bit in size. The size of the input fed is din to be 64 bits input is fed to the present encryption stage module and further these aids in the encryption of the input data. The updated Ku_output (28 bit) from the encryption module is forwarded to the decryption module. Also, the hardware architecture of PRESENT encryption and decryption 128bit Key block cipher follows the design discussed in 80bit key cipher model.

4. RESULTS AND DISCUSSION

The system design is performed by using Verilog code and implemented on Artix-7 FPGA board of device 7A100T-3 CSG324. The verification of the designed system is verified by using a design supporting tool like Xilinx 14.7 and is simulated by using modelsim 6.3f. The execution of the system gives the significant results respective to outputs. The RTL of the proposed PRESENT cipher model is given in Figure.5, which contains 64bit data input (din), 80bit key_input, clock (clk), d_load, k_load and yields corresponding output of 64bit (dout).

The simulation results obtained from the PRESENT ciper model (80bit key) is given in Figure.6. If the 31st cycle reaches high as k_load then reaches to low in next cycle. Later, d_load is extended to logic high and the encryption of data input (din) process begins with the parallel key updation process. Once the key updation process is completed, the obtained data output (dout) is generated on simulation results.

Similarly, the PRESENT cipher model (128bit key) RTL diagram is shown in Figure.7, which consists of 64bit datainput (din), 128bits key input, clock (clk), d_load and k_load. The corresponding outputs to the inputs are 64bit d_out and 5bits r_count.

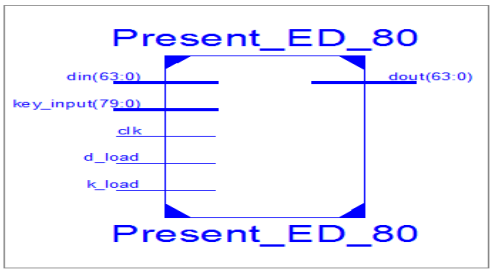

Figure 5 Top module of PRESENT cipher model (80bit key)

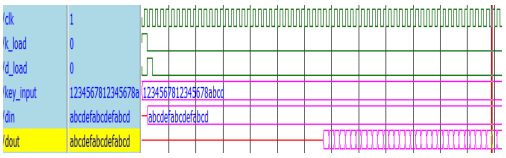

Figure 6 Simulation results of (80bit key) PRESENT cipher model

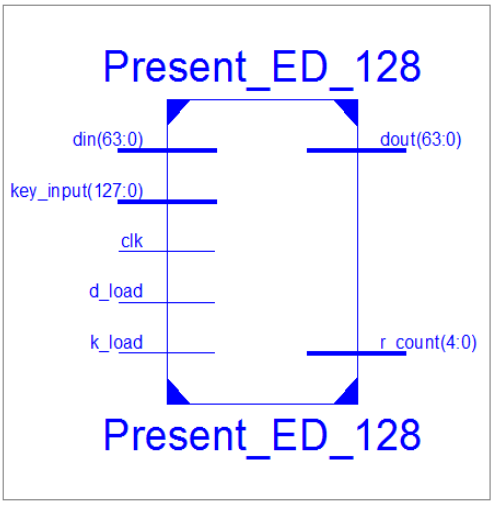

Figure 7 Top module of PRESENT cipher model (128bit key)

The simulation results obtained from the PRESENT cipher model (128bit key) is given in Figure.8. If the 31st cycle reaches high as k_load then reaches to low in next cycle. Later, d_load is extended to logic high and the encryption of data input (din) process begins with

the parallel key updation process. Once the key updation process is completed, the obtained data output (dout) is generated on simulation results.

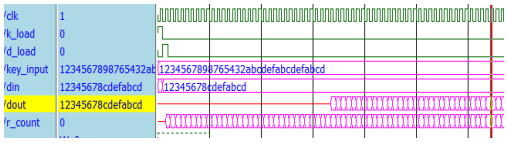

Figure 8 Present-Encryption-Decryption 128-bit key Simulation Results

The performance analysis of the proposed cipher model is performed by comparing with Sbeiti et al. [20] parameters such as count, FF's, maximum frequency, throughput, slices, clock cycles, number of keys and LUT's. The target device utilized for comparison with Sbeiti et al. [20], and proposed PRESENT cipher model is Spartan-3X3S400-3FG464. The comparative results are shown in Table.3.

Table 3 Comparative results of 80bit and 128bits keys

Parameter	Encryption				Decryption			
	80bit		128bit		80bit		128bit	
	Sbeiti et al. [20]	Proposed	Sbeiti et al. [20]	Proposed	Sbeiti et al. [20]	Proposed	Sbeiti et al. [20]	Proposed
State	64	64	64	64	64	64	64	64
Key	80	80	128	128	80	80	128	128
LUT's	350	290	300	213	328	250	366	299
FF's	154	183	200	229	154	149	202	197
Slices	202	181	202	158	197	162	221	191
Fmax (MHz)	240	196	254	196	232	239	239	239
Clock Cycles	32	32	32	32	32	32	32	32
Throughput	480	392	508	338	476	478	478	478

5. CONCLUSION

This paper introduces the hardware architecture implementation of PRESENT cipher model using 80bit and 128bit key encryption and decryption blocks. The design of PRESENT cipher model is performed by using Verilog programming language on Xilinx ISE 14.7 platform implemented over Artix-7 FPGA for both encryption and decryption through 80bit and 128bit key modules. In the proposed PRESENT cipher model, a 64bit data path enables the entire round execution in a single cycle. This execution operation requires 64bits permutation layer as well as sixteen S-box layers of 4bits. Thus, to perform the key scheduling two different sized key were selected and designed the 80bit and 128bit keys. The proposed PRESENT cipher model also involves the decryption process to acquire the original input data. In this paper, both the 80bit and 128 bit key based PRESENT cipher model are compared to analyse the performance at both encryption and decryption by considering slices , LUT's, Flip-flops (FF's), Frequency and throughput as performance parameters. From the analysis it is found that throughput is improved with PRESENT cipher model. The proposed PRESENT cipher model can be further considered with other Cryptography algorithm to enhance the security of hybrid algorithm for more data of consumer-side applications.

REFERENCES

[1] Usman, Muhammad, et al. "Sit: A lightweight encryption algorithm for secure internet of things." arXiv preprint arXiv:1704.08688, 2017

[2] Xu, Teng, James B. Wendt, and Miodrag Potkonjak. "Security of IoT systems: Design challenges and opportunities." Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. IEEE Press, 2014.

[3] Katagi, Masanobu, and Shiho Moriai. "Lightweight cryptography for the internet of things." Sony Corporation, pp. 7-10, 2008

[4] McKay, Kerry A., et al. Report on lightweight cryptography. US Department of Commerce, National Institute of Standards and Technology, 2017.

[5] Yalla, Panasayya, and Jens-Peter Kaps. "Lightweight cryptography for FPGAs." Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on. IEEE, 2009.

[6] Banik, Subhadeep, et al. "Midori: A block cipher for low energy." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014.

[7] Karri, Ramesh, Grigori Kuznetsov, and Michael Goessel. "Parity-based concurrent error detection of substitution-permutation network block ciphers." International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2003.

[8] De Canniere, Christophe, Orr Dunkelman, and Miroslav Knežević. "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers." Cryptographic Hardware and Embedded Systems-CHES Springer, Berlin, Heidelberg, 272-288, 2009

[9] Standaert, François-Xavier, et al. "SEA: A scalable encryption algorithm for small embedded applications." International Conference on Smart Card Research and Advanced Applications. Springer, Berlin, Heidelberg, 2006.

[10] Guo, Xu, Zhimin Chen, and Patrick Schaumont. "Energy and performance evaluation of an

FPGA-based SoC platform with AES and PRESENT coprocessors." International Workshop on Embedded Computer Systems. Springer, Berlin, Heidelberg, 2008.

[11] Sbeiti, Mohamad, et al. "Design space exploration of present implementations for fpgas." Programmable Logic, 2009. SPL. 5th Southern Conference on. IEEE, 2009.

[12] Yalla, Panasayya, and Jens-Peter Kaps. "Lightweight cryptography for FPGAs." Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on. IEEE, 2009.

[13] Duan, Xiaoyi, et al. "Differential power analysis attack and efficient countermeasures on PRESENT." Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on. IEEE, 2016.

[14] Guo, Xiaofei, et al. "Simulation and analysis of negative-bias temperature instability aging on power analysis attacks." Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on. IEEE, 2015.

[15] Rahaman, Hafizur, Jimson Mathew, and Dhiraj K. Pradhan. "Secure testable s-box architecture for cryptographic hardware implementation." The Computer Journal 53.5 (2010): 581-591.

[16] Kavun, Elif Bilge, and Tolga Yalcin. "RAM-based ultra-lightweight FPGA implementation of PRESENT." Reconfigurable Computing and FPGAs (ReConFig), 2011 International Conference on. IEEE, 2011.

[17] Tay, J. J., et al. "Compact FPGA implementation of PRESENT with Boolean S-Box." Quality Electronic Design (ASQED), 2015 6th Asia Symposium on. IEEE, 2015.

[18] Lara-Nino, Carlos Andres, Miguel Morales-Sandoval, and Arturo Diaz-Perez. "Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher." Digital System Design (DSD), 2016 Euromicro Conference on. IEEE, 2016.

[19] Chouhan, Tushar Singh. "Implementation of Present Cryptographical Algorithm for the Encryption of Messages in NETFPGA 1G." Computational Intelligence and Communication Networks (CICN), 2015 International Conference on. IEEE, 2015.

[20] Sbeiti, Mohamad, et al. "Design space exploration of present implementations for FPGAS." Programmable Logic, 2009. SPL. 5th Southern Conference on. IEEE, 2009.