International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Lightweight Structure of Random Key Generation for PRESENT Block Cipher

Bahram Rashidi

Faculty of Engineering, Ayatollah Boroujerdi University, Boroujerd, Iran.
Corresponding author's email: b.rashidi@abru.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 04- December-2023
Received in revised form: 29-Jan-2023
Accepted: 22-Feb-2024
Published online: 10-March-2024

Keywords:
PRESENT block cipher,
Random key generation,
Lightweight,
High-throughput.

ABSTRACT

In this paper, we design a lightweight and modified random key generation for PRESENT block cipher which is applicable in the encryption of the digital signals. In the block ciphers, the master key is used directly in the encryption process for the data (plaintext). But in this work, a master key (initial key) is used to derive the new random master keys (random session keys) and use these keys for the encryption process. The use of random keys will overcome the brute force attack that can be applied to the PRESENT cipher. The random session keys generated will produce different ciphertexts for the same plaintext for every session. In this approach, we take advantage of the block cipher to produce random keys. The PRESENT cipher is shared in both random key generation and encryption process. Therefore, the proposed structure has both random key generation and data encryption in a unified circuit. This property reduces hardware resources. The implementation results, in 180 nm CMOS technologies, show the proposed structure is comparable in terms of area and delay with other works.

I. Introduction

Due to the rapid advancements in communication systems and digital broadcasting, data security in these systems has become a major research challenge. It is essential to develop techniques for providing security. To secure the proprietary, digital signals such as image, sound, ... need to be ciphered before transmission using encryption techniques. The security of encrypted digital signals can always be improved through new encryption methods. Therefore, new encryption schemes that can protect data are efficiently researched. In recent years, many cryptographic techniques such as lightweight block ciphers have been proposed for the security of the digital signals [1]-[2]-[3]. PRESENT [4] is a lightweight block cipher that is standardized in the ISO/IEC 29192-2, with an efficient structure [5]. This cipher is suitable for the realization of crypto-processors. It has the Substitution Permutation Network (SPN) structure with 64-bit block size and 80- and 128-bit key sizes. The number of rounds in the PRESENT cipher is equal to 31. Add round key, substitution layer (S-box), and permutation layer are the main blocks of the PRESENT cipher. In work [6] the identification and classification of recent research practices about the flexible hardware implementation of cryptographic algorithms are presented. The identified researches have been classified according to three design approaches: (1) cryptoprocessor, (2) crypto coprocessor, and (3) multicore crypto processor. Consequently, a comparative analysis of various cryptographic algorithms in terms of flexibility, throughput, area, power, and implementation technology has been presented. Based on work [11] the PRESENT cipher has a reasonable area and time complexities for cryptography applications. This block cipher is suitable for low-cost and ultra-light implementations. The throughput and speed of this block cipher are also important

factors for hardware implementation. Hardware implementation of a cryptographic system has advantages over software implementation, such as increasing the speed of data processing and increasing the security of that system. On the other hand, random key generation is a very important issue for symmetric key encryption. Therefore, in this paper, we are looking for the optimal hardware implementation of a cryptographic system with the random key generation ability.

The hardware design of block ciphers is an important subject in the literature. The area consumption and time delay of the symmetric key cryptosystems depend on the block ciphers. Therefore, a block cipher is a key subject in determining implementation performance. Several hardware structures of the PRESENT cipher have been reported in works [7]-[16]. In work [8] three structures of the PRESENT consisting of pipelined structure, serial structure, and round structure are proposed. Between these structures, the pipelined structure has the most area compared to the other structures. The serial structure is the slowest compared to the other structures. The round structure has more throughput compared to the serial structure, but it consumes a high area. The PRESENT algorithm is implemented based on a Single-cycle structure in work [9]. An optimized circuit for the S-box is presented in work [10]. In work [11] a low latency and high throughput structure of the PRESENT is proposed based on the loop unrolling technique. Also, the S-box is implemented based on a low-area circuit. In work [12] both key sizes 80- and 128-bit are supported based on a high-throughput and flexible hardware structure of the PRESENT algorithm for IoT applications. In [15] the design of three different types of S-box architectures for the PRESENT cipher to optimize the design parameters for resource-constrained applications are presented. In the previous works, there is no random key generation unit. This subject is the main limitation of existing circuits for PRESENT cipher. In addition, the works [9] and [12] consume high hardware resources for the implementation of the PRESENT cipher. On the other hand, the computation time and the number of clock cycles for generating ciphertext in the works [7]-[8] and [16] are also high. In the present work, we have achieved acceptable hardware results and time specifications compared to previous works. Also, in the proposed system, there is a random key generation unit based on the PRESENT algorithm is used to reduce the hardware.

The encryption of digital signals such as image encryption is needed to perform real-time communication with a random key generator unit. So how to carry out the image encryption has also become a hot issue [17]-[20]. The key generator can improve the security issues of the encryption algorithms and increase the amount of text required for the differential attacks [21]-[22]. Therefore, in this paper, we present a modified random key generation algorithm for the PRESENT cipher. The results, in 180 nm CMOS technology, show that the proposed structure has acceptable hardware resources, timing characteristics, and security properties compared to the other works. The contributions of this paper are as follows:

- In this work, the master key is used to derive the new random master keys (random session keys) and use these keys for the encryption process. In this approach, we take advantage of encryption to produce random keys.
- The PRESENT cipher is shared in both random key generation and encryption processes. This property reduces hardware resources.
- The proposed structure has both random key generation and data encryption in a unified system which can be used in digital signal encryption with a high number of data.
- In the PRESENT cipher, to further reduce the logic gates of the 4-bit S-boxes, we applied further simplifications on the expression terms of the S-boxes for more area optimization. Therefore, a low-cost 4-bit S-box for the PRESENT cipher is achieved. For the 128-bit key, the area consumption, computation time, and throughput of the proposed method are equal to 2583 GEs, 52.928 ns, and 1209 Mbps, respectively, which are acceptable compared to the other works.

The rest of the paper is organized as follows. The PRESENT cipher is summarized in Section 2. The random key generation method is presented in Section 3. In Section 4 the proposed structure of random key generation is described. Security of the proposed structure is presented in Section 5. Section 6 shows a comparison between our structure and related works. Finally, the paper is concluded in section 7.

II. PRESENT Block Cipher

The PRESENT is a 31-round block cipher for low-cost cryptographic applications [4]. In each round, we have a 64-bit round key addition, 16 4-bit substitution boxes (S-boxes), and a 64-bit permutation layer (pLayer). It has a block size of 64-bit and a key size of 80- or 128-bit. The round keys are the 64 most significant bits of the supplied key.

A. S-box

The substitution layer is composed of 16 4×4-bit S-boxes. The S-box used in PRESENT is a 4-bit to 4-bit S-box. The input nibbles (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) are substituted by the output nibbles (C, 5, 6, B, 9, 0, A, D, 3, E, F, 8, 4, 7, 1, 2), respectively.

B. Proposed Protection Scheme Using DS-DOCRs Considering N-1 Contingency

The PRESENT can support keys with sizes 80- or 128-bit. The key schedule of the PRESENT cipher consists of a left cyclic shift (rotate to left), several S-boxes, and a 5-bit XOR

with the values of the least significant bit of the round counter (RC). Let the main key be $K = k_{127}, k_{126}, \ldots, k_1, k_0$ ($K = k_{79}, k_{78}, ..., k_1, k_0$ for 80-bit). In round i, the 64-bit round key $K_i = \kappa_{63}, \kappa_{62}, ..., \kappa_1, \kappa_0$ consists of the 64 leftmost bits of the current K content. Thus, in round i we have $K_i = \kappa_{63}, \kappa_{62}, \ldots, \kappa_1, \kappa_0 = k_{127}, k_{126}, ... k_{65}, k_{64}$ and $K_i = \kappa_{63}, \kappa_{62}, \ldots, \kappa_1, \kappa_0 = k_{79}, k_{78}, \ldots, k_{17}, k_{16}$ for 80-bit. The key update process for 80-bit keys is as follows:

$$\kappa_{i+1}[79:76] = S - box(\kappa_i[79:76]), \quad (1)$$

$$\kappa_{i+1}[75:20]||\kappa_i[14:0] = \kappa_i[75:20]||\kappa_i[14:0], \quad (2)$$

$$\kappa_{i+1}[19:15] = \kappa_i[19:15] \oplus RC. \quad (3)$$

Also, for the 128-bit keys we have the following key update:

$$\kappa_{i+1}[127:124] = S - box(\kappa_i[127:124]), \quad (4)$$

$$\kappa_{i+1}[123:120] = S - box(\kappa_i[123:120]), \quad (5)$$

$$\kappa_{i+1}[119:67]||\kappa_i[66:0] = \kappa_i[119:67]||\kappa_i[66:0], \quad (6)$$

$$\kappa_{i+1}[66:62] = \kappa_i[66:62] \oplus RC. \quad (7)$$

The operations $\oplus$ and $||$ represent bit-wise XOR and concatenation, respectively.

III. Random Key Generation

The random key generation which is used in this paper is based on work [23]. Here, we modify the random key generation algorithm of work [23] for the PRESENT cipher. The random session keys are generated based on generated random numbers. In the block ciphers, the master key is used directly in the encryption process. But in this work, a master key is used to derive the new random master keys (random session keys) and use these keys for the encryption process. In this method, we take advantage of the encryption process to produce random numbers. The PRESENT cipher is used as the heart of the random number generation. To our knowledge, this paper is the first work to use the PRESENT cipher for the random key generation process. Because this process generates strong random numbers as the random keys. We take advantage of the block cipher to produce random keys. The proposed structure has both random key generation and data encryption in a united circuit. This block cipher is suitable for low-cost and ultra-light implementations. The throughput and speed of this block cipher are also important factors for hardware implementation. Therefore, the PRESENT cipher is shared in both random key generation and encryption processes. This feature reduces hardware resources and the generation of strong random numbers.

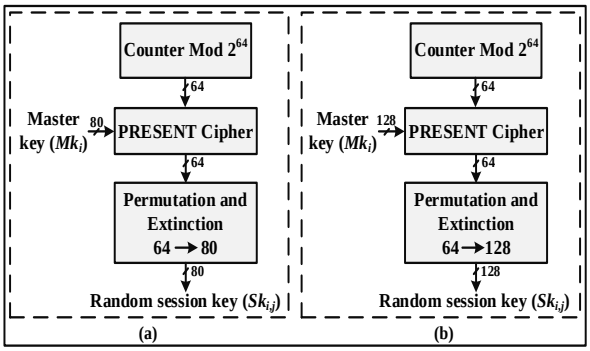


Fig. 1. Random key generation based on the PRESENT cipher for 80-bit key (a) and 128-bit key (b).

A counter with period $N = 2^{64}$ produces input to the encryption as plaintext. This counter is called the modulus counter (MOD counter). It is defined based on the number of states that the counter will sequence through before returning to its original value (0 value). In the random key generation part, a 64-bit counter that counts from 0 to $2^{64} - 1$ in decimal, has a modulus value of $N = 2^{64}$ so would therefore be called a modulo-N, or mod-N, counter. Note also that it has taken N clock cycles to get from 0 to $2^{64} - 1$ values. The session keys are generated from the master keys Km_i as the main key, where i is a natural number and the counter values as plaintexts. It can generate 2^{64} random session keys from each master key. After each session key is produced, the counter is incremented by one. Random key generators based on the PRESENT cipher for 80-bit keys and 128-bit keys are shown in Figs. 1 (a) and (b), respectively.

The hierarchy of key generation from the master key to round keys is summarized in Fig. 2. In this method, we can generate many new master keys by using a secure approach based on the used block cipher in the system. Therefore, the need for the generation of the master keys is simply provided. In brute force attacks, the attacker applies different combinations of keys to hack the system. But this fracture leads to confusion and diffusion. In the PRESENT cipher, the 80- and 128-bit key lengths are to be produced. The proposed procedure of random key generation based on the PRESENT cipher for the 80- and 128-bit key sizes are shown in Algorithms 2 and 3, respectively. In this algorithm, we generate random keys based on PRESENT block cipher as the heart of the process, MOD counter, and expansion/permutation unit. Each of the session key outputs $Sk_{i,j}$ is based on a different counter value and therefore for $i = 1$, we have $Sk_{1,0} \neq Sk_{1,1} \neq Sk_{1,1} \neq \ldots \neq Sk_{1,2^{64}-1}$. It is said that if the cryptographic system (ciphertext) does not have enough details to find plaintext, it is a secure cryptographic system. Because the master key is protected, it is not computable to accurately deduce any of the secret keys through knowledge of one or earlier keys. In this case, the same plaintext can create

Algorithm 1 Random key generation based on PRESENT cipher for 80-bit keys

Input: Counter values Cv_j and the 80-bit master key Mk_i.
Output: Random Session Key $Sk_{i,j}$.
1. **For** i **from** 0 **to** Mkn **do** // The number of Mkn values depends on the number of master keys.
2. **For** j **from** 0 **to** Kn **do** // The maximum value of Kn is $2^{64}-1$.
3. $X = Addkey(Cv_j, Mk_i)$; // Start of the PRESENT encryption.
4. **For** r **from** 2 **to** 32 **do**
5. $Y = S-box(X)$;
6. $W = Permutation(Y)$;
7. $X = Addkey(W, K_r)$; // The K_r are the round keys that are generated from Cv_j by the key scheduling.
8. **End For**;// End of the PRESENT encryption.
9. $Pe = PE_{64\to80}(X)$ // The $PE_{64\to80}(X)$ is permutation and expansion from 64-bit to 80-bit.
10. $Sk_{i,j} = Pe$;
11. **End For**;
12. **End For**;

Algorithm 2 Random key generation based on PRESENT cipher for 128-bit keys

Input: Counter values Cv_j and the 128-bit master key Mk_i.
Output: Random Session Key $Sk_{i,j}$.
1. **For** i **from** 0 **to** Mkn **do** // The number of Mkn values depends on the number of master keys.
2. **For** j **from** 0 **to** Kn **do** // The maximum value of Kn is $2^{64}-1$.
3. $X = Addkey(Cv_j, Mk_i)$; // Start of the PRESENT encryption.
4. **For** r **from** 2 **to** 32 **do**
5. $Y = S-box(X)$;
6. $W = Permutation(Y)$;
7. $X = Addkey(W, K_r)$; // The K_r are the round keys that are generated from Cv_j by the key scheduling.
8. **End For**;// End of the PRESENT encryption.
9. $Pe = PE_{64\to128}(X)$ // The $PE_{64\to128}(X)$ is permutation and expansion from 64-bit to 128-bit.
10. $Sk_{i,j} = Pe$;
11. **End For**;
12. **End For**;

different ciphertexts using random session keys. Fig. 3 shows the generation of different ciphertexts from the same plaintext using random session keys. The plaintext P_0 for master key Mk_0 with session keys $Sk_{0,0}$, $Sk_{0,1}$, $Sk_{0,2}$, ..., $Sk_{0,2^p-1}$ generate the ciphertexts C_0, C_1, C_2, ..., C_{2^p-1}, where p is the size of plaintext. Therefore, brute force attacks will not be able to discover the key.

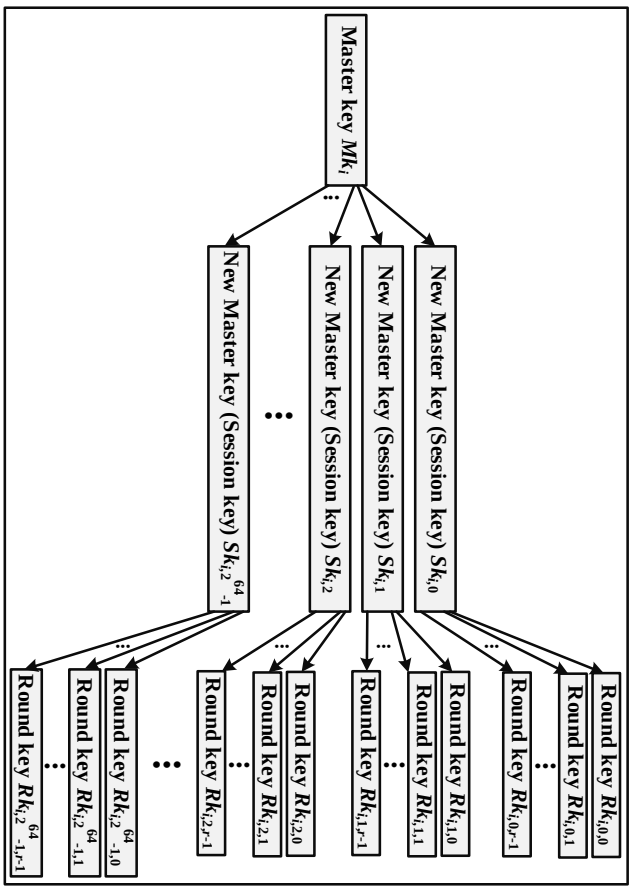


Fig. 2. Hierarchy of key generation from the master key to round keys.

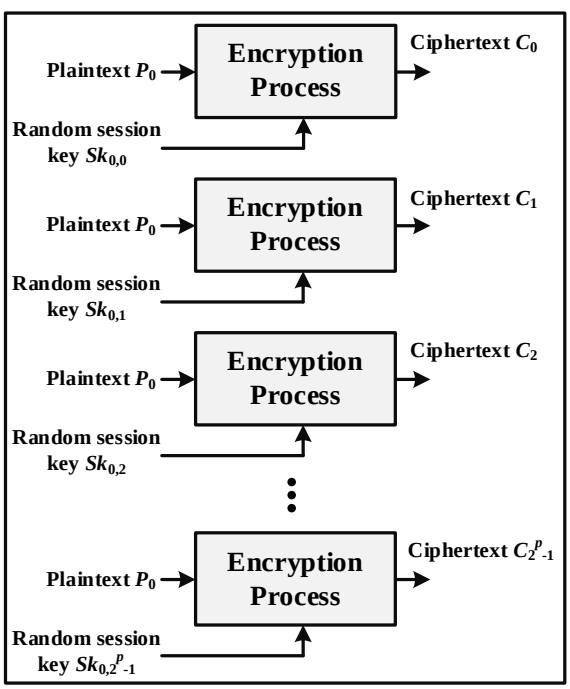


Fig. 3. Generating different ciphertexts from the same plaintext using random session keys.

IV. Proposed Structure of PRESENT Cipher with Random Key Generation

In this section, we present the proposed structure of the PRESENT cipher with random key generation. The reduction of hardware resources for cryptosystems has been achieved in different ways. In the proposed structure, the two methods are used to reduce the hardware resources. These methods include 1- The PRESENT cipher is shared in both random key generation and encryption processes. This property reduces hardware resources. 2- In the PRESENT cipher, to further reduce the logic gates of the 4-bit S-boxes, we applied further simplifications on the expression terms of the S-boxes for more area optimization. Therefore, a low-cost 4-bit S-box for the PRESENT cipher is achieved.

The proposed structure is shown in Fig. 4. The structure is constructed based on round function and key scheduling of PRESENT (includes main blocks such as 16 S-boxes, permutation layer, two registers called Reg $_r$, Reg $_k$), 64-bit counter, permutation and expansion block, and several 2-to-1 multiplexers. This circuit can perform two operating modes (dual-mode circuit). The first mode is the random key generation with RK_ENC='0' and in the second mode encryption operation is performed with RK_ENC='1'. Therefore, in the proposed structure, the PRESENT cipher as the main core is shared between the two modes of the random key generation and the encryption process. In this case, the hardware resource is reduced. We have two procedures for the use of the proposed structure. In the first procedure (Procedure 1) after generating each random session key, we encrypt the input data (plaintext) to generate the ciphertext based on this random session key. In the second procedure (Procedure 2) after generating the all required random session keys and storing these keys in memory we start the encryption process for generating ciphertext based on these stored random session keys.

For generating the first random key, at the first clock cycle, the control signals RK_ENC and Start are set to '0' and '1', respectively, and the first value of the counter (0), as plaintext, and the master key, as the main key, are applied to the structure. In the next clock cycles, the control signal Start is set to '0', and the round computations of the PRESENT are processed. At the end of round computations, the output of register Reg $_r$ (ciphertext) is applied to the permutation and expansion block for generating the first random session key. For producing the second random key, the MOD counter is increased by one and the procedure is similar to the first random key. Also, in the encryption process, at the first clock cycle, the control signals RK_ENC and Start are set to '1'. In this step, in the Procedure 1, the generated random key which is produced in the previous mode (random key generation) is used as a master key for the encryption of plaintext (Procedure 1).

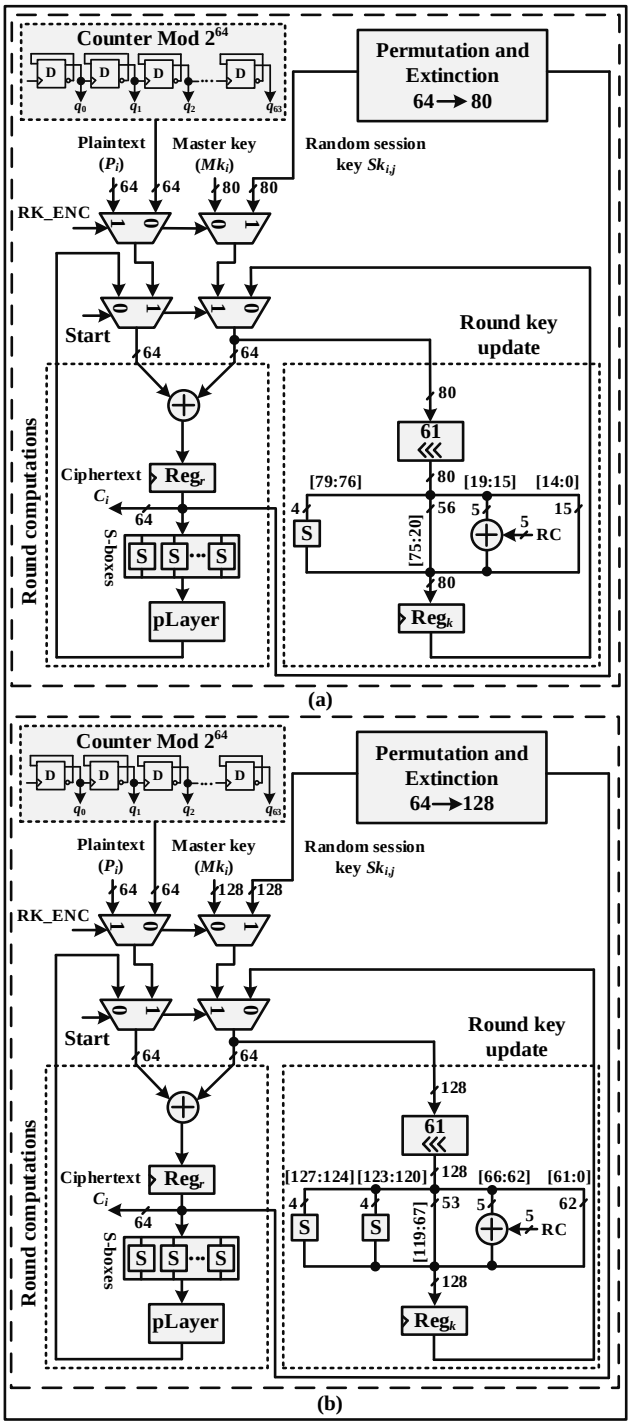


Fig. 4. Proposed structure of PRESENT cipher with random key generation.

After completion of the round computations, the output of register Reg $_r$ is considered as ciphertext. For Procedure 2, the random key (as a master key) can be applied from the memory.

A. Permutation and Expansion

Since the ciphertext in the PRESENT cipher is a 64-bit value and the random session keys that are generated based on the structure have 80- and 128-bit sizes. Therefore, we need the permutation and expansion units for conversion of

TABLE 1: THE BIT PERMUTATION AND EXPANSION $PE_{64\to80}(x)$.

i	0	1	2	3	4	5	6	7
$P(i)$	0	8	16	32	48	1	9	17
i	16	17	18	19	20	21	22	23
$P(i)$	11	19	35	51	4	12	20	36
i	32	33	34	35	36	37	38	39
$P(i)$	22	38	54	7	15	23	39	55
i	48	49	50	51	52	53	54	55
$P(i)$	41	57	10	18	26	42	58	11
i	64	65	66	67	68	69	70	71
$P(i)$	60	13	21	29	45	61	14	22
i	8	9	10	11	12	13	14	15
$P(i)$	33	49	2	10	18	34	50	3
i	24	25	26	27	28	29	30	31
$P(i)$	52	5	13	21	37	53	6	14
i	40	41	42	43	44	45	46	47
$P(i)$	8	16	24	40	56	2	17	25
i	56	57	58	59	60	61	62	63
$P(i)$	19	27	43	59	12	20	28	44
i	72	73	74	75	76	77	78	79
$P(i)$	30	46	62	15	23	31	47	63

64-bit to 80- and 128-bit. The permutation and expansion is an operation for conversion of 64-bit ciphertext of the PRESENT cipher to 80- and 128-bit, for the two key sizes 80- and 128-bit. Therefore, we present two permutation and expansion operations $PE_{64\to80}(x)$ and $PE_{64\to128}(x)$ for the 80- and 128-bit key sizes, respectively. The permutation and expansion operation for 80-bit key size transforms bit x of ciphertext to bit position $PE_{64\to80}(x)$ as follows:

$$PE_{64\to80}(x) = \begin{cases} 0 & x = 0, \\ 8 & x = 1, \\ 16 & x = 2, \\ 32 & x = 3, \\ 64 & x = 4, \\ PE_{64\to80}(x-5)+1 & 5 \le x \le 79. \end{cases} \quad (8)$$

Also, for the generating 128-bit key size, we have the permutation and expansion $PE_{64\to128}(x)$ as follows:

$$PE_{64\to128}(x) = \begin{cases} 0 & x = 0, \\ 12 & x = 1, \\ 18 & x = 2, \\ 24 & x = 3, \\ 30 & x = 4, \\ 36 & x = 5, \\ 42 & x = 6, \\ 48 & x = 7, \\ PE_{64\to128}(x-8)+1 & 8 \le x \le 127. \end{cases} \quad (9)$$

The details of the $PE_{64\to80}(x)$ is given in Table 1.

Proposed structure of 4-bit S-box

The PRESENT S-box compared to other 4-bit S-boxes has a suitable security level for cryptographic applications [24]. In this paper, a low-cost 4-bit S-box for the PRESENT cipher is achieved. The low-cost S-boxes are applicable for area-constrained cryptography applications. It is designed based on a simple combinational logic with acceptable area and delay results. As mentioned before, the main and complex block in the PRESENT cipher is the S-box. It has a key role in the performance of the hardware structure. A low-area approach for the PRESENT S-box is presented in work [25]. The computation of S-box are as follows [25]:

$T_1 = x_2 \oplus x_1,\ T_2 = T_1 x_1,\ T_3 = x_0 \oplus T_2,\ y_3 = x_3 \oplus T_3,\ T_2 = T_1 T_3,\ T_1 = T_1 \oplus Sb_3,\ T_2 = T_2 \oplus x_1,\ T_4 = x_3 + T_2,\ y_2 = T_1 \oplus T_4,\ T_2 = T_2 \oplus x_3,,\ y_0 = y_2 \oplus T_2,\ T_2 = T_2 \oplus T_1,\ y_1 = T_3 \oplus T_2.$

In these equations, the input bits and the output bits are denoted as x_3, x_2, x_1, x_0 and y_3, y_2, y_1, y_0, respectively. In the following, we present an optimized version of these equations:

$$y_3 = x_3 \oplus T_3 = x_3 \oplus x_0 \oplus (x_1(x_2 \oplus x_1)) = x_3 \oplus x_0 \oplus (x_1 x_2,). \quad (10)$$

In the y_2 equation, the terms 1 T_2 and 2 T_1 can be rewritten as follows:

1 $T_2 = (x_2 \oplus x_1)(x_0 \oplus (x_1 x_2,)) \oplus x_1 = ((x_2 x_1, + x_2, x_1)(x_0 \oplus x_1 x_2,) \oplus x_1) + (x_1,(x_0 x_1, x_2 + x_0 x_2 x_1, + x_2, x_1 x_0,)) = x_2 x_1, x_0 + x_1(x_2 + x_0) = x_2(x_0 + x_1) + x_1 x_0.$

2 $T_1 = x_3 \oplus x_2 \oplus x_1 \oplus x_0 \oplus x_1 x_2, = x_3 \oplus x_2 \oplus x_1 x_2 \oplus x_0 = x_3 \oplus x_2 x_1, \oplus x_0.$

In this case, the y_2, y_1, and y_0 equations are present as follows:

$$y_2 = (x_3 + T_2) \oplus T_1 \quad (11)$$

$$y_1 = T_2 \oplus T_1 \oplus x_0 \oplus x_1 x_2 \quad (12)$$

$$y_0 = y_2 \oplus T_2 \oplus x_3 \quad (13)$$

The proposed structure of the PRESENT S-box is shown in Fig. 5. The S-box is implemented using 7 XOR, 3 AND, 4 OR, and 2 NOT gates. Therefore, it is implemented by only 16 logic gates. The critical path delay of the S-box in the structure is equal to $4T_X$, where T_X is the time delay of the 2-input XOR gate. Table 2 shows the results of the proposed structure and other works. The area and delay of

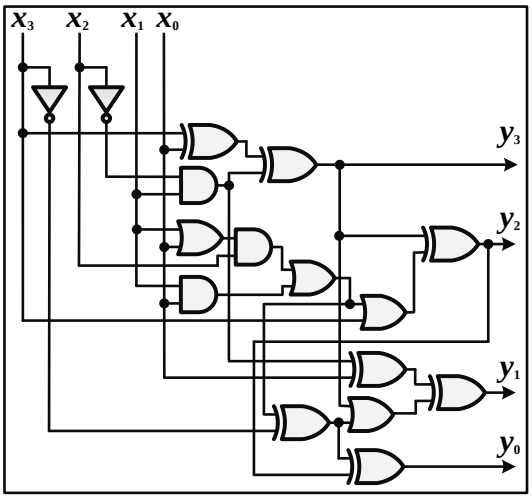


Fig. 5. Optimized structure of 4-bit S-box in the PRESENT cipher.

TABLE 2: RESULTS OF THE PRESENT S-BOX FOR THE PROPOSED STRUCTURE AND OTHER WORKS.

Works	# AND (OR OR)	# NAND (or NOR)	# XOR (or XNOR)	CPD
[26]	43	—	—	$2T_A+3T_O$
[11]	20	—	7	$T_X+T_A+2T_O$
[12]	2	1	10	$6T_X+2T_A$
[13]	39	8NOT	—	$3T_O+T_A+T_N$
[14]	7+8AND3	—	23	$7T_X+3T_A+2T_{A3}$
[15]	28	—	10	$T_X+T_A+3T_O$
TW	7	—	7	$4T_X$

TW: This work; CPD: Critical path delay; T_X, T_A, T_O, T_{A3} are the time delay of the 2-input XOR gate, 2-input AND gate, 2-input OR gate, and 3-input AND gate, respectively.

PRESENT S-box for this work and recent work [12] are equal to (22 GEs and 0.56 ns) and (24 GEs and 1.086 ns), respectively, using Synopsys Design Compiler with 180 nm CMOS technology. As seen from this table, the proposed method has acceptable area and delay parameters.

B. Application of the proposed structure

The proposed structure is more suitable for the encryption of a large amount of digital data. The application domains for the proposed structure of key generation for the PRESENT cipher include image encryption (medical images, industrial images, fingerprint images, ...), voice encryption, and any area that needs to be protected from security breaches. In recent years, more digital images have been transmitted through networks, and most of them have to be transmitted through public networks. To transmit hidden digital images to receivers, digital image encryption technology must be used. Over the past decades, many image encryption algorithms have been proposed [3]. The

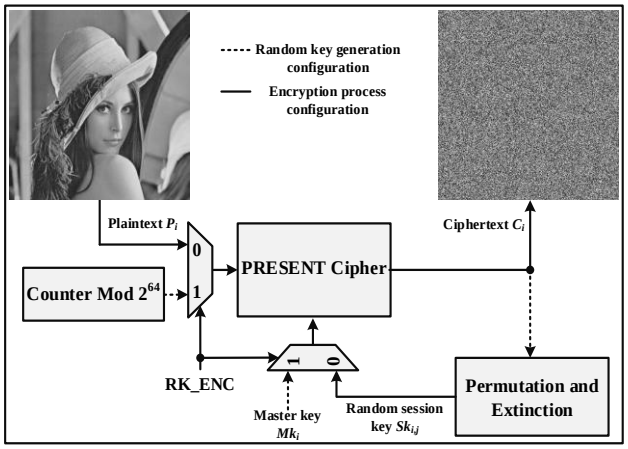


Fig. 6. Configuration of the proposed structure for image encryption.

configuration of the proposed structure for image encryption is shown in Fig. 6. As mentioned before, the system has two configurations including random key generation and encryption process, which are controlled based on the control signal RK_ENC. For data encryption, it is necessary to generate many random session keys, so the proposed structure is suitable for cryptographic applications with a high number of data.

V. Security of the Proposed Structure

The protection of private keys is crucial in private key cryptography, as any disclosure of these keys can be used to decrypt secret messages. To improve the security of private keys, we propose a key generation algorithm that generates the random private key of a user that meets the current security requirements of any private key algorithm. In the proposed structure, the required private random keys are generated by the system itself. This eliminates the system's need to achieve the keys from outside. This protects the produced keys. This is while the previous implementations have not the key generation unit and require receiving the key from outside the system, which reduces the security of the keys.

The main focus of this work is the design and implementation of a lightweight structure of random key generation for the PRESENT block cipher. However, we analyze the security of structures from a hardware point of view. Side-channel attacks are the biggest threats to the security of cryptographic algorithms. These attacks are used to recover sensitive data such as the main key and plaintext. Side-channel attacks on cryptographic devices are non-invasive passive attacks that use certain physical information leaked during normal encryption such as power consumption [27], time delay [28], or electromagnetic radiation [29] to find the secret key. Simple Power Analysis is a method that interprets power consumption

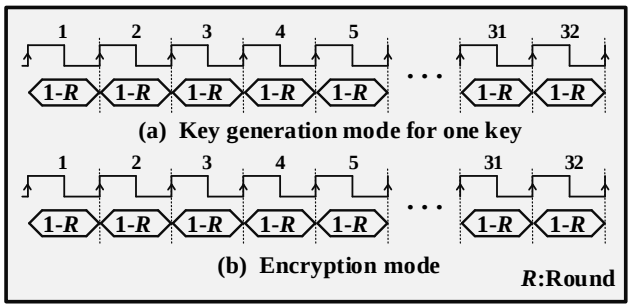


Fig. 7. The computations of key generation mode (a) and encryption mode (b) at each clock cycle.

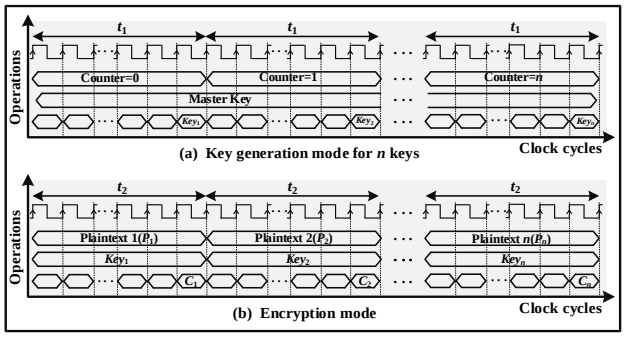


Fig. 8. Waveform of the proposed structure for the computation of key generation and encryption of the plaintexts with the same execution time (t_1=t_2).

measurements during cryptographic operations. It can achieve information about a device's operation as well as the secret key (or plaintext) based on a power trace.

A. Power analysis of the proposed hardware structures

In the proposed structures, at each clock cycle, we have the computation of operations with a similar hardware complexity. In this case, the power consumption at each clock cycle is almost constant. In the proposed structures, the same operations are performed at each clock cycle. For example, at each clock cycle, for both the key generation mode and the encryption mode, we have the computation of a round of the PRESENT cipher. In each round, the computation of S-boxes and permutation layer is performed. Figs. 7 (a) and (b) show the computations of key generation mode and encryption mode, respectively, at each clock cycle. As seen from these figures, at each clock cycle the computations are similar (one round (1-R) at one clock cycle). In this case, the power consumption at each clock cycle is fixed. Therefore, this feature leads to a unified power trace in total clock cycles and the power traces are independent of the key and plaintext message patterns.

B. Power analysis of the proposed hardware structures

The timing attack is another important side-channel attack. In this attack, the time taken to execute a key generation or encryption of plaintext is measured precisely by the attacker [28]. If the execution time for different plaintexts is different, this will lead to the attacker obtaining information about the bit-pattern of a plaintext. Therefore, the implementation of the algorithm must reduce data-dependent timing information. The computation time for each key generation operation or encryption of a plaintext in the proposed structure is fixed. In this case, the structures leak no information about the bit-pattern of the plaintext (or key) bit-pattern. The computation of a random key and encryption of a plaintext takes the same time t_1 and t_2, respectively. Fig. 8 shows the waveform of the proposed structure for a random key and encryption of a plaintext. As seen in this figure, a random key generation takes the same time t_1, and also the encryption of a plaintext takes the same time t_2. On the other hand, because both computations are implemented based on PRESENT cipher, so the time t_1 and t_2 are equal. Therefore, the computation time of the proposed architecture is independent of the key and plaintext being manipulated. In this case, the details of the internal computations of the key generation and plaintext algorithms are hidden.

VI. Results and Comparison

The hardware complexity of the proposed structure of the PRESENT cipher with random key generation and other works are compared in this section. We use the Synopsys Design Compiler tool based on the library of standard cells with 180 nm CMOS technology to achieve the ASIC results. The area and critical path delay, number of clock cycles, throughput, and throughput/area ratio parameters are used for the evaluation of performance. The performance evaluation in terms of throughput/area is useful, where both the constraints of throughput and area are required to be fulfilled at the same time in many cryptographic applications. Table 3 shows the hardware results for the proposed structure and other implementations of the PRESENT cipher. In the other works, available in the literature, only the PRESENT cipher is implemented and the previous structures lack a random key generation unit. But in the proposed work, we have implemented the PRESENT cipher with a random key generation unit. Therefore, this property must be considered in the comparison. Because, in the proposed structure, the parameters such as the number of clock cycles (CCs), time, and throughput are computed for the generation of a random key and data encryption (in other words, it is assumed that a random key is first generated, and subsequently, a plaintext is encrypted using this key to produce ciphertext). But in the other works, these parameters are computed for only data encryption without the random key generation.

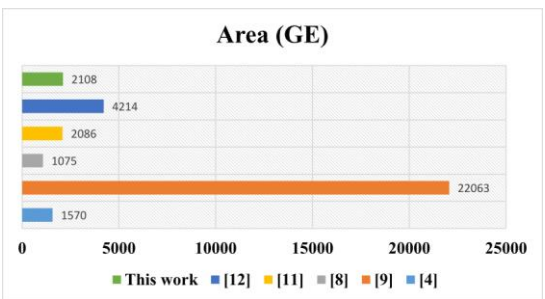
In work [8] three structures of the PRESENT consisting of pipelined structure, serial structure, and round structure are proposed. Between these structures, the pipelined structure has the most area compared to the other structures. The serial structure is the slowest compared to the other

TABLE 3: RESULTS OF THE PROPOSED IMPLEMENTATION AND OTHER WORKS ON THE .PRESENT CIPHER.

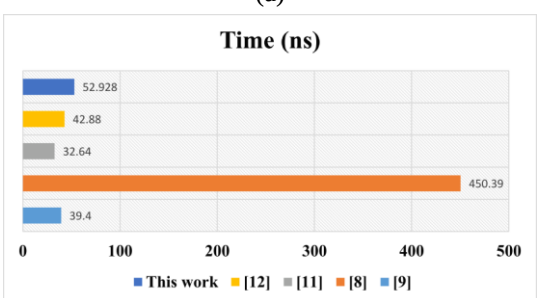
Works	TECHNOLOGY	Area (GE)	#CCs	CPD (ns)	Time (ns)	Thr. (Mbps)	Thr./Area Mbps/GE
[4] K-128	180 nm	1886	—	—	—	—	—
[4] K-80	180 nm	1570	—	—	—	—	—
[9] K-128	180 nm	23005.75	1	38.10	38.10	1,570	0.068
[9] K-80	180 nm	22063.50	1	39.40	39.40	1,510	0.068
[8] S K-80	180 nm	1075	563	0.80	450.39	142.10	0.132
[8] S1 K-128	180 nm	2989	40	3.09	120.98	529	0.177
[8] S2 K-128	180 nm	2900	63	2.83	178.27	359	0.124
[7] S K-128	180 nm	1296	563	2.89	1,627.07	39.33	0.030
[11] K-128, UF=1	180 nm	2305.75	32	1.02	32.64	1,961	0.851
[11] K-80, UF=1	180 nm	2086.30	32	1.02	32.64	1,961	0.940
[12] K-64	180 nm	4214	32	1.34	42.88	1492.54	0.354
[12] K-128	180 nm	4214	32	1.34	42.88	1492.54	0.354
[16] K-64	180 nm	1098	622	—	—	—	—
[16] K-128	180 nm	1879	250	—	—	—	—
TW, K-80	180 nm	2108	64	0.827	52.928	1,209	0.574
TW, K-128	180 nm	2583	64	0.827	52.928	1,209	0.468

TW: This work; S:Serial; GE: Gate equivalents; UF: Unroll factor; K: Keys; CCs: Clock cycles; Thr.: Throughput.

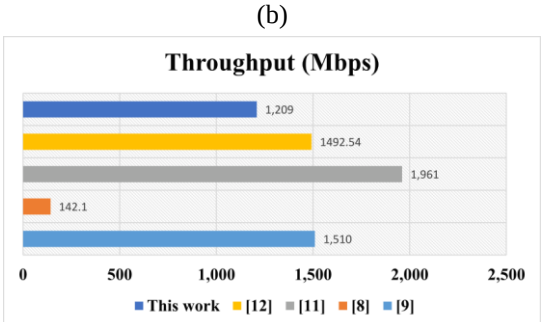
structures. The PRESENT algorithm is implemented based on a Single-cycle structure in work [9]. An optimized circuit for the S-box is presented in work [10]. In work [11] a low latency and high throughput structure of the PRESENT is proposed based on the loop unrolling technique. Also, the S-box is implemented based on a low-area circuit. In work [12] both key sizes 80- and 128-bit are supported based on a high-throughput and flexible hardware structure of the PRESENT algorithm for IoT applications. As seen from the table, the proposed structure has a reasonable implementation cost. This structure can be a good candidate for image encryption with low area consumption and an acceptable security level. We implemented the proposed structure with the 80-bit key, which resulted in 2108 gates with a throughput/area of 0.574 Mbps/GE. This corresponds to 2583 gates with a maximum delay of 0.468 Mbps/GE for the 128-bit key. Figs. 9 (a), (b), (c), and (d) show column diagrams of the area, execution time, throughput, and throughput/area, respectively, for the proposed structure and other works for 80-bit key size. Also, for the case key size of 128-bit, these parameters are shown in Fig. 10. Based on the hardware results, we get acceptable improvement in terms of throughput/area for the PRESENT block cipher with key generation unit.



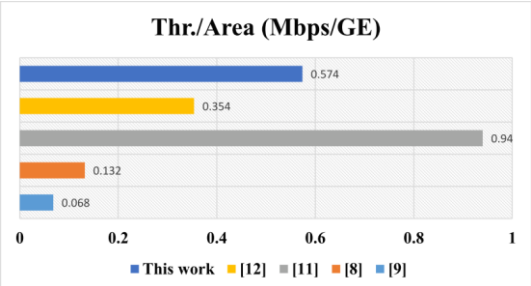
(a)



(b)

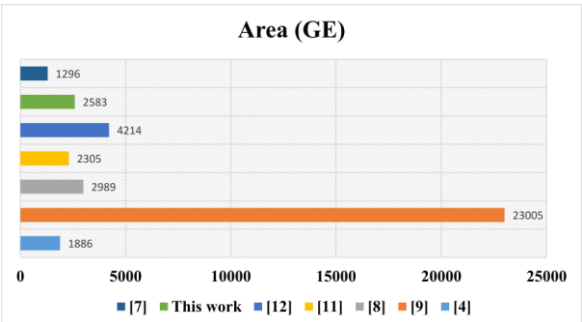


(c)

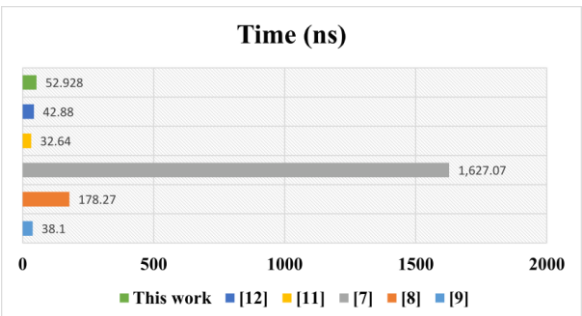


(d)

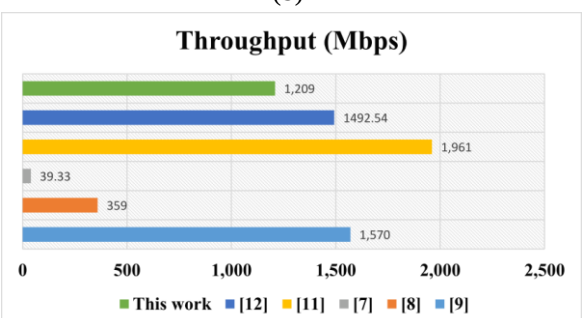
Fig. 9. Column diagrams of the Area (a), Time (b), Throughput (c), Throughput / Area (d) for the proposed structure and other works for 80-bit key size.



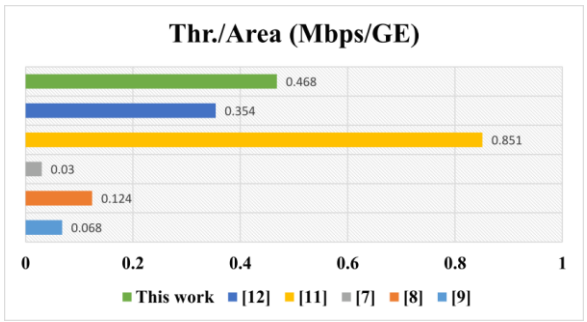
(a)



(b)



(c)



(d)

Fig. 10. Column diagrams of the Area (a), Time (b), Throughput (c), Throughput / Area (d) for the proposed structure and other works for 128-bit key size.

VII. Conclusions

One of the important issues in many block ciphers is random key generation, especially in the encryption of a high number of digital signals. The use of random keys will overcome the brute force attack that can be applied to a block cipher. In this paper, we design a hardware structure of a modified random key generation for lightweight PRESENT block cipher which is applicable in encryption of the digital signal. In this work, a master key is used to derive the new random master keys (random session keys) and use these keys for the encryption processing. We take advantage of the block cipher to produce random keys. The proposed structure has both random key generation and data encryption in a unified circuit. Therefore, the PRESENT cipher is shared in both random key generation and encryption process. This feature reduces hardware resources. The implementation results, in 180 nm CMOS technologies, show the proposed structure has a suitable area and throughput compared to other works.

References

[1] Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I. and Manifavas, C., A review of lightweight block ciphers, *Journal of Cryptographic Engineering*, Vol. 11, Iss. 3, 2018, pp. 141-184.

[2] Sadhukhan, R., Patranabis, S., Ghoshal, A., Mukhopadhyay, D., Saraswat, V. and Ghosh, S., An Evaluation of Lightweight Block Ciphers for Resource-Constrained Applications: Area, Performance, and Security, *Journal of Hardware and Systems Security*, Vol. 1, Iss. 3, 2017, pp. 203-218.

[3] You, L., Yang, E., and Wang, G., A novel parallel image encryption algorithm based on hybrid chaotic maps with OpenCL implementation, *Soft Computing*, Vol. 24, 2020, pp. 12413-12427.

[4] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin Y. and Vikkelsoe, C., PRESENT: An ultra lightweight block cipher, *in Proc. Cryptographic Hardware and Embedded Systems-CHES, Springer*, 2007, Vienna, Austria, pp. 450-466.

[5] International Standardization of Organization (ISO):

Information Technology-Security Techniques-Lightweight Cryptography-Part 2: Block Ciphers, document ISO/IEC 29192-2, Jan. 2012.

[6] Rashid, M., Imran, M., Jafri, A.R., Al-Somani, T.F., Flexible Architectures for Cryptographic Algorithms-A Systematic Literature Review, *Journal of Circuits, Systems, and Computers*, Vol. 24, No. 3, 2018, pp. 1-32.

[7] Wang, C., and Heys, H.M., An ultra compact block cipher for serialized architecture implementations, *in Proc. Canadian Conference on Electrical and Computer Engineering*, 2009, St. John's, NL, Canada, pp. 1-6.

[8] Rolfes, C., Poschmann, A., Leander, G., Paar, C., Ultra-Lightweight Implementations for Smart Devices-Security for 1000 Gate Equivalents, *in Proc. International Conference on Smart Card Research and Advanced Applications, Springer*, 2008, London, UK, pp. 89-103.

[9] Maene, P., and Verbauwhede, I., PRESENT: An ultra lightweight block cipher, *in Proc. International Workshop on Lightweight Cryptography for Security and Privacy*, 2015, Vol.9542, Bochum, Germany, pp. 131-147.

[10] Rekha, S.S., and Saravanan, P., Low Cost Circuit Level Implementation of PRESENT-80 S-BOX, *in Proc. International Symposium on VLSI Design and Test, Springer*, 2017, Roorkee, India, pp. 354-362.

[11] Rashidi, B., Efficient and High-throughput ASIC Implementations of HIGHT and PRESENT Block Ciphers, *IET Circuits, Devices & Systems*, 2019, Vol. 13, Iss. 6, pp. 731-740.

[12] Rashidi, B., Flexible Structures of Lightweight Block Ciphers PRESENT, SIMON and LED, *IET Circuits, Devices & Systems*, 2020, Vol. 14, Iss. 3, pp. 369-380.

[13] Sherine Jenny, R., Sudhakar, R., Karthikpriya, K. Design of Compact S Box for Resource Constrained Applications, *Journal of Physics: Conference Series*, 2021, Vol. 1767, pp. 1-12.

[14] Panchami, V., Mary Mathews, M., A Substitution Box for Lightweight Ciphers to Secure Internet of Things, *Journal of King Saud University-Computer and Information Sciences*, 2023, Vol. 35, pp. 75-89.

[15] Mishra, R., Okade, M., Mahapatra, K., Optimized S-Box Architectures of PRESENT Cipher for Resource Constrained Applications, *in Proc. IEEE International Symposium on Smart Electronic Systems*, 2020, Chennai, India, pp. 1-4.

[16] Parthasarathy, P., Saravanan, Efficient Hardware Implementation of PRESENT Lightweight Cipher, *in Proc. International Conference on Intelligent Systems for Communication, IoT and Security*, 2023, Coimbatore, India, pp. 1-6.

[17] N.Noura, H., Chehab, A., Raphael, C. Efficient & secure cipher scheme with dynamic key-dependent mode of operation, *Signal Processing: Image Communication*, 2019, Vol. 78, pp. 448-464.

[18] Ismail Abdelfatah, R. Secure Image Transmission Using Chaotic-Enhanced Elliptic Curve Cryptography, *IEEE Access*, 2019, Vol. 8, pp. 3875-3890.

[19] Shanthakumari, R. and Malliga, S., Dual layer security of data using LSB inversion image steganography with elliptic curve cryptography encryption algorithm, *Multimedia Tools and Applications*, 2020, Vol. 79, pp. 3975-3991.

[20] Yang, C.H., Wu, H.C., and Su, S.F., Implementation of Encryption Algorithm and Wireless Image Transmission System on FPGA, *IEEE Access*, 2019, Vol. 7, pp. 50513-50523.

[21] Penchalaiah, P., and Ramesh Reddy, K., Random multiple key streams for encryption with added CBC mode of operation, *Perspectives in Science*, 2016, Vol. 8, pp. 57-60.

[22] Montero-Canela, R., Zambrano-Serrano, E., Tamariz-Flores, E.I., Munoz-Pacheco, J.M., and Torrealba-Melendez, R., Fractional chaos based-cryptosystem for generating encryption keys in Ad Hoc networks, *Ad Hoc Networks*, 2020, Vol. 97, pp. 1-21.

[23] Pradeep, L.N., and Bhattacharjya, A., Random Key and Key Dependent S-box Generation for AES Cipher to Overcome Known Attacks, *in Proc. International Symposium on Security in Computing and Communication, Springer*, 2013, Mysore, India, pp. 63-69.

[24] Rashidi, B., Lightweight Cryptographic S-Boxes Based on Efficient Hardware Structures for Block Ciphers, *The ISC International Journal of Information Security*, Vol. 15, Iss. 1, 2022, pp. 137-151.

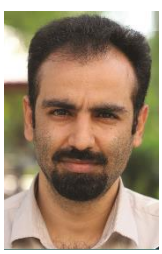
[25] Courtois, N.T., Hulme, D., and Mourouzis, T., Solving Circuit Optimisation Problems in Cryptography and Cryptanalysis, *in Proc. the fifth workshop on Special-Purpose Hardware for Attacking Cryptographic Systems*, Washington, DC, USA, 2012, pp. 179-191.

[26] Tay, J.J., Wong, M.L.D., Wong, M.M., Zhang, C. and Hijazin, I., Compact FPGA implementation of PRESENT with Boolean S-Box, *in Proc.* 6th *Asia Symp. Quality Electron. Design*, Aug. 2015, pp. 144-148.

[27] Kocher, P., Jaffe, J., and Jun, B., Differential power analysis, *in Proc. of Advances in Cryptology*, 1999, Berlin, Germany, pp. 388-397.

[28] Kocher, P.C., Timing Attacks on Implementations of Diffie"Hellman, RSA, DSS, *in Proc. of Advances in Cryptology*, 1996, Berlin, Germany, pp. 104-113.

[29] Hayashi, Y.I., and Homma, N., Mizuki, T., Aoki, T., Sone, H., Sauvage, L., and Danger, J.L., Analysis of electromagnetic information leakage from cryptographic devices with different physical structures, *IEEE Transactions on Electromagnetic Compatibility*, Vol. 55, No. 3, 2013, pp. 571-580.



Bahram Rashidi was born in Boroujerd, Iran, in 1986. He received his B.S. degree in electrical engineering from Lorestan University, Iran, in 2009 and he received his M.S. from Tabriz University, Iran in 2011 also he obtained his Ph.D degree from Isfahan University of Technology (IUT), in 2016, where he is currently an associate professor in the department of electrical engineering at University of Ayatollah Boroujerdi. His research interests include hardware implementation for the arithmetic of finite fields, cryptographic hardware, Block ciphers and VLSI circuits for elliptic curve cryptosystems.