Circuits Syst Signal Process (2018) 37:1116–1133
https://doi.org/10.1007/s00034-017-0592-8

Programmable Cellular Automata-Based Low-Power Architecture to S-Box : An Application to WBAN

Bhoopal Rao Gangadari[1] · **Shaik Rafi Ahamed**[1]

Received: 20 May 2016 / Revised: 17 June 2017 / Accepted: 20 June 2017 /
Published online: 30 June 2017
© Springer Science+Business Media, LLC 2017

Abstract In this paper, we presented a low-power, less area architecture of S-Box used in advanced encryption standard (AES) using programmable cellular automata (PCA). The proposed architecture performance in terms of security is evaluated using cryptographic properties such as nonlinearity, input/output entropy, correlation immunity bias, balancedness property, strict avalanche criterion, and it is found that the proposed method is secure enough for cryptography applications. The proposed architecture of AES with PCA-based S-Box is implemented on ASIC using TSMC 0.18-μm and UMC 0.13-μm CMOS technology libraries. Simulation studies show that the proposed architecture has average energy consumption of 58.702 nJ, power dissipation of 3.259 mW, area of 0.184 mm^2, for TSMC 0.18-μm at 13.69 MHz and energy consumption of 18.275 nJ, power dissipation of 1.026 mW, area of 0.069 μm^2 for UMC 0.13-μm at 13.69 MHz. The proposed architecture shows reduction in power dissipation by 83% and in energy consumption by 10% compared to the best classical S-Box and composite field arithmetic-based S-Box for AES algorithm. The S-Box using PCA is more flexible and dynamic in nature with low power, lesser energy consumption area and hence suitable for wireless body area network applications.

Keywords Advanced encryption standards (AES) · Application specific integrated circuit (ASIC) · Substitution Box (S-Box) · Cellular automata (CA) · Programmable cellular automata (PCA) · Wireless body area network (WBAN)

✉ Bhoopal Rao Gangadari
bhoopal@iitg.ernet.in

Shaik Rafi Ahamed
rafiahamed@iitg.ernet.in

[1] Department of EEE, Indian Institute of Technology Guwahati, Guwahati, India

1 Introduction

Cryptography plays an important role in the area of secure communication systems. In order to protect the data, encryption algorithms like AES, Camellia, are used in cryptography for the purpose of security. AES is a symmetric key algorithm of Federal Information Processing Standards (FIPS) Publication 197, issued as a replacement of Data Encryption Standards (DES) [21] by the National Institute of Standards and Technology (NIST) in 2001 [1]. AES was standardized and adopted in the latest IEEE Standard 802.15.6 for wireless body area network (WBAN) due to high security, efficiency, ease of implementation, higher data rates [10]. Moreover, this algorithm is widely used in applications like secure communication, RFID tags.

The AES algorithm was realized on hardware using pipelining, sub-pipelining and loop unrolling architecture to achieve maximum throughout. The architecture of AES algorithm was implemented on 0.18-μm CMOS ASIC technology using fully pipelining for an encryption process, achieving a throughput of 30–70 Gbits/s [13]. Although these architectures were efficient for many applications which require high throughput. Moreover, these high-throughput architecture hardware realizations utilize more area and high power consumption. Among these architectures, the hardware implementation of classical S-Box was traditionally designed using LUT's [19,24]. In order to enhance the speed and avoid unbreakable delay, the S-Boxes are also designed and implemented using composite field arithmetic, which involves in decomposition of Galois Field GF(2^8) to GF($(2^4)^2$) or GF($(((2^2)^2)^2$), respectively, using isomorphic mapping [15,28]. The S-Box was realized using binary decision diagram (BDD), and TBoxes provided a throughput of 10 Gbps in the literature [2,9]. FPGA-based implementation of AES processor is reported in [18]. The works so far reported in the literature mainly emphasized on enhancement in throughput and reduction in hardware complexities [11]. However, there is a necessity to develop an alternative architecture which is secure enough with a lesser area and low energy consumption. Since the WBAN applications demand an ultralow power to increase the lifetime of the battery, we proposed the PCA-based S-Box realization, which enormously reduces power consumption compared to conventional LUT-based S-Box realization. In order to check the level of security for the proposed PCA-based S-Box realization, we also obtained cryptographic properties such as nonlinearity, entropy, correlation immunity bias, balancedness property and strict avalanche criterion. It is also found that the proposed PCA-based S-Box gives comparable performance in terms of security with respect to LUT-based S-Box (Table 4).

In this paper, the concept of AES algorithm is revisited in Sect. 2. The formulation of S-Box using cellular automata is discussed in Sect. 3. The proposed novel PCA-based dynamic S-Box and architecture are presented in Sect. 4. The comparative analysis of LUT-based S-Box and PCA-based S-Box is evaluated using cryptographic properties with architecture implementation in Sect. 5, and conclusion is drawn in Sect. 6.

2 Concept of AES Algorithm

The AES algorithm is a symmetric key cryptographic algorithm as shown in Fig. 1, which uses four transformations in each round, namely the substitution bytes (S-Box),

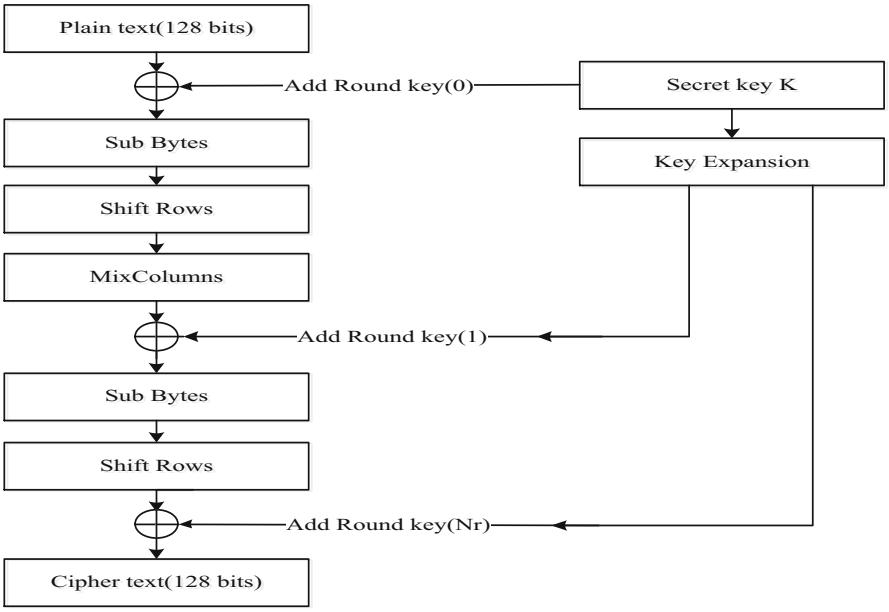

Fig. 1 A block diagram of AES encryption

shift rows (SR), mix columns (MC) and add round key (ARK), to generate cipher text over plain text in order to provide the desired level of security.

The rounds of transformation (N_r) used in the AES algorithm can be determined using the relation $N_r = \frac{S_k}{32} + 6$, where S_k = key size. For wireless body area network (WBAN) application, the latest IEEE Standard 802.15.6 has recommended a secret key size of 128 bits for encryption and decryption, which results in 10 rounds of transformations [10].

The initial add round key is the 128-bit direct secret key which is used in EXOR operation with the input data, and subsequent add round keys are generated in the key expansion phase. Out of these 10 rounds, the first 9 rounds undergo all the four transformations, whereas the last round performs only the three transformations S-Box, SR and ARK as illustrated in Fig. 1. In each round of encryption process, the algorithm performs S-Box, SR, MC and ARK operation on a 4×4 array of bytes called as a state as described in the following subsections.

2.1 Substitution Bytes

In S-Box transformation, each byte of the input state is substituted by another byte using a precomputed lookup table (LUT). The S-Box is computed by multiplicative inverse over the Galois finite field GF (2^8), using the irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$, followed by an affine transformation. Mathematically, the affine transformation of S-Box in matrix is as follows:

Table 1 LUT-based S-Box

y																
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fd	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{1}$$

Traditionally, the classical S-Box will be implemented using memory cells which can store the 256 possible values in 8×8 array of bits. For input data of 128 bit, a total number of sixteen LUT-based S-Boxes are required for AES algorithm. The LUT-based S-Box in hexadecimal form is represented in Table 1. For example, if the input is $a5$, then the substituted value of S-Box is determined from Table 1 by the intersection of a row and 5 column which results in 06.

2.2 Shift Rows

In SR transformation, the first row remains unchanged and the subsequent three rows are shifted cyclically to the left by 1, 2 and 3 bytes, respectively. This transformation is attained in order to create diffusion in cipher text.

2.3 Mix Columns

The MC transformation operates column wise, where in each column four term polynomials over GF(2^8) and multiplied by a modulo x^4+1 with a fixed polynomial $A(x) = (03H)x^3 + (01H)x^2 + (01H)x + (02H)$. Mathematically, this operations can be written in matrix form as follows:

$$S^1(x) = A(X) \otimes S(x). \tag{2}$$

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02H & 03H & 01H & 01H \\ 01H & 02H & 03H & 01H \\ 01H & 01H & 02H & 03H \\ 03H & 01H & 01H & 02H \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \tag{3}$$

where $0 \leq C < 4$.

2.4 Add Round Key

In ARK transformation, each round key consists of 4-byte words denoted by w_i which are generated from the key expansion. Key expansion block generates a total of 4(N_r+1) number of ARKs. In this AES algorithm initial phase, the first round key is the initial 128 bits of secret key and the subsequent round keys are calculated iteratively using SubWord, RotWord and Rcon. Each ARK is 4-word output from the key expansion block denoted by ARK $i = (w_{4i},\ w_{4i+1},\ w_{4i+2},\ w_{4i+3})$, where $i = 0\ to\ N_r$. SubWord means nonlinear transformation of each byte of key using S-Box. The rotation word (RotWord) is a cyclic left shift of each byte in a word by one byte. Rcon is an array of constant words, and the left most byte in a word is nonzero.

3 Reformulation of S-Box Using Cellular Automata

The basic function of S-Box is to transform one byte of input data into another one byte secret data using predefined lookup table (LUT). The truth table of S-Box is basically a function $f : B^n \rightarrow B^m$.

The LUT-based S-Box architecture requires more area and consumes high energy. Hence, LUT-based S-Box architecture is not suitable for IEEE Standard 802.15.6 for WBAN applications. Moreover, IEEE Standard 802.15.6 for WBAN also demands a highly secure, lesser area and low energy consumption cryptographic algorithm. In order to meet the requirements of WBAN, in this paper, we proposed a cellular automata (CA)-based architecture for realization of S-Box.

The basic structure of CA is shown in Fig. 2, which consist a groups of cells with a finite size of length from R_0 to R_7 which evolve at discrete time steps using deterministic rule with each cell storing one of the two states 0 and 1. If the right most and left most (extreme) cells of this finite size CA are considered to be adjacent each other, then the CA is called as periodic boundary CA. The one-dimensional periodic

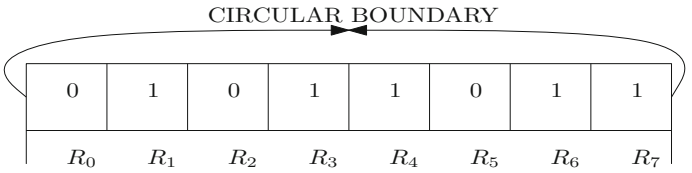

Fig. 2 A cellular automata array of size ($R_0 - R_7$) with a circular boundary condition

Table 2 Truth table for rule 90 and 75

	111	110	101	100	011	010	001	000	
Decimal 90	0	1	0	1	1	0	1	1	Rule 90
Decimal 150	0	1	0	0	1	0	1	1	Rule 75

boundary CA evolves with different neighborhood configurations of elementary CA. Each elementary CA consists of central cell i which is surrounded by neighborhood cells of a defined radius r; therefore, the total number of cells in elementary CA is given as $n_i = 2r + 1$, including central cell i. We considered $r = 1$, and the total number of possible different neighborhood configurations of elementary CA are $L = 2^{n_i}$ with R_{i-1}^t, R_i^t, R_{i+1}^t number of cells. The next state central cell R_i^{t+1} at a time step $(t+1)$ depends on the current state of central cell R_i^t and also neighborhood R_{i-1}^t, R_{i+1}^t cells, respectively, at time t with a deterministic rule of function f_p. Mathematically, R_i^{t+1} can be expressed as

$$R_i^{t+1} = f_p(R_{i-1}^t, R_i^t, R_{i+1}^t) \tag{4}$$

The representation of deterministic rules f_p in decimal form is shown in Table 2, and the total number of CA rules considered is $2^L = 256$. If the rule in CA is expressed using EXOR logic and/or EXNOR logic, then it is called as additive CA. The additive CA is used in VLSI testing, bit-error correcting code and data encryption. If all the cells in CA evolve using the same deterministic rule, then the CA is called uniform CA. The dynamic nature of one-dimensional periodic uniform CA depends on deterministic rule f_p and the number of iterations. In this paper, we considered a programmable cellular automata (PCA) which is a modified version of one-dimensional periodic uniform CA structure in which all the cells in the lattice obey the same rule [26].

The functioning of PCA-based S-Box with 256 number of different rules is shown in Algorithm 1. There exist a relationship between the number of iterations at discrete time step t and group of CA cells R_0 to R_7 in a lattice, as diversification from input to output is high if the time step $t \geq$ size of the lattice. In CA algorithm, D is loaded with 8-bit initial random value, so there exists 2^8 possible random initial states which are taken into consideration. However, the 8 bit random initial states of PCA evolves using different 256 deterministic rule and NOI means the number of iterations which varies from time step 1 to 50.

Algorithm 1 Programmable Cellular Automata with 256 rules

$R \leftarrow 0$;
$R^t \leftarrow D$;
$INPUT \leftarrow f_p$;
$INPUT \leftarrow NOI$;
loop $r \leftarrow 1$ *to* 256
  for $t \leq NOI$ **do**
    if $Rule(r) == f_p$ **then**
      for $i \leftarrow R_0$ *to* R_7 **do**
        $R_i^{t+1} \leftarrow f_p(R_{i-1}^t, R_i^t, R_{i+1}^t)$
      end for
    end if
  end for
end loop

4 Proposed PCA-Based S-Box

The S-Box of AES algorithm in cryptography provides confusion in the cipher text and hence plays a important role in AES algorithm. The conventional LUT-based S-Box realization uses a large number of memory cells which eventually consumes more power. Moreover, the secret information from the existing AES algorithm architecture can be revealed using power analysis attacks [23].

In order to overcome these limitations, we proposed a PCA-based architecture for S-Box with low energy consumption and dynamic in nature. Unlike the conventional LUT-based S-Box, the proposed S-Box is dynamic in nature because of the fact that the output of the S-Box is a function of input rule which can be programmed. Out of 10 rounds of transformations, each encryption round in AES algorithm as discussed in Sect. 2 consists of substitution bytes (S-Box), shift rows (SR), mix columns (MC) and add round key (ARK) transformations. The substitution bytes block transformation in each round of encryption is replaced with the proposed PCA-based S-Box as shown in Fig. 3. The detailed description of proposed PCA-based S-Box is presented in the following paragraphs.

There are a total number of 256 rules that can be used to program into the registers at discrete time steps. The output R_i^{t+1} depends on the input control signals R_{i-1}^t, R_i^t, R_{i+1}^t as shown in Fig. 4. For example, in Table 1, if 90 is the input rule and 110 is input data , the output R_i^{t+1} should be 1.

The output of the proposed basic PCA structure with given 8-bit input rule is one bit as shown in Fig. 4. In order to implement the S-Box which operates on 8 bits, eight such basic cells shown in Fig. 4 need to be interconnected.

The proposed architectural design of 8×8 array PCA-based S-Box is implemented using logic gates, multiplexers and registers as shown in Fig. 5. The initial 8 bits of CA array will be loaded into register R_1 using preset and clear signals. The bits in the register R_1 will be applied as control signals to 8:1 MUX (M_1–M_8) in circular fashion whose input is an 8-bit rule. First 3 bits R_7, R_0 and R_1 will act as a control signals to M_1, and the rotated bits R_0, R_1 and R_2 to M_2 and the last MUX M_8 the control signal are R_6, R_7 and R_0. The MUXs produce the output which is shown in Table 1.

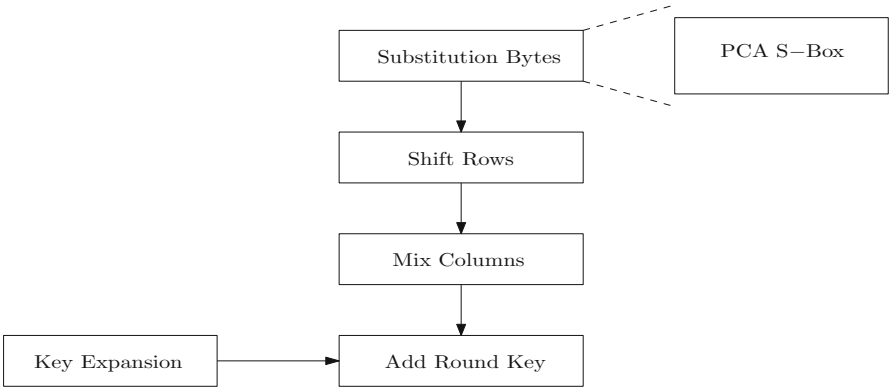

Fig. 3 Encryption process of each round

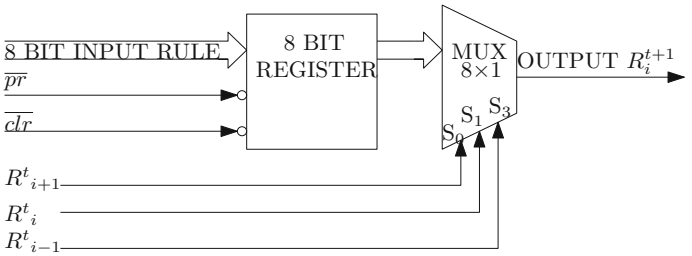

Fig. 4 PCA basic cell structure

The multiplexer outputs so produced will be used as a CA array bits in subsequent iterations.

The control logic has a 6-bit upcounter and a comparator. If the count value of counter is equal to the number of iteration in time step, then the output of the control logic circuit goes high to enable the register (R_2). The latency incurred in computing the S-Box depends upon the number of iterations of PCA. However, on the other side, the ASIC implementation of PCA-based S-Box architecture shown in Fig. 5 utilizes few logic elements compared to that of LUT-based S-Box [14]. As a result, CA-based S-Box architecture consumes less power and require small chip area, and hence, this hardware realization is much suitable for WBAN applications.

5 Performance Comparison Between Conventional LUT S-Box and Dynamic PCA S-Box

In order to analyze the security aspects, the output bits obtained by the proposed RCA^2-based S-Box architecture as described in Sect. 4 are taken as inputs bits to the MATLAB system which computes cryptographic properties. However, in order to examine the S-Box using cryptographic properties the 2^8 output bits are transformed into a single output bit using Boolean function $f_i : B^n \rightarrow B$, where $i \in (1, m)$. In a S-Box,

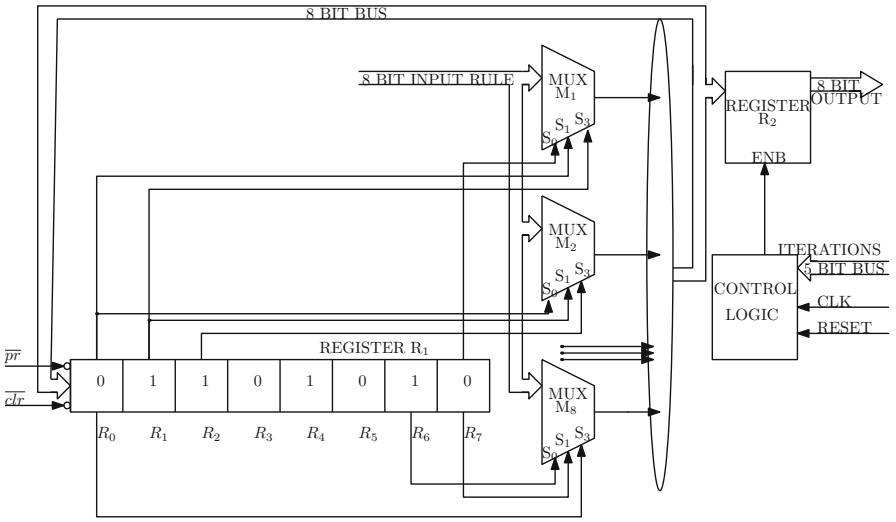

Fig. 5 Proposed PCA-based S-Box

Table 3 Symbolic representation

Cryptographic property	Notation
Strict avalanche criteria(SAC)	Υ_S
Nonlinearity	$\Re_S$
Entropy	H_S
Correlation immunity bias (CIB)	$\Phi_S(l)$

$f : B^n \rightarrow B^m$, and hence, there exists m number of functions $\mu = \{f_1, f_2, \ldots, f_m\}$. The truth table representation of S-Box in polarity form is written as $f_k(x) = (-1)^{f(x)}$.

$$f_\beta(x) = (\alpha_1 f_1(x) \oplus \alpha_2 f_2(x) \oplus \alpha_3 f_3(x) \ldots \oplus \alpha_m f_m(x)) \tag{5}$$

The Boolean function f_β is a linear combination of m functions $f_i(x), i \leq m$, where $\alpha_i \in B^m$ are coefficients of the linear function.

5.1 Analysis Using Cryptographic Properties

The level of security provided by PCA-based S-Box is analyzed using cryptographic properties for 256 different rules at discrete time step t which varies from 8 to 50. It is also observed that the diversification in a output pattern of PCA-based S-Box is high if the number of iterations used in order to evolve CA is not less than the number of cells in a lattice ($t \geq K$) [3]. The symbolic representation of the cryptographic properties is presented in Table 3.

5.1.1 Balancedness Property

A Boolean function is balanced, if number of 1s is equal to number of 0s. The balancedness of the output is measured by Hamming weight. If the Hamming weight is 2^{n-1}, then the output is balanced. This property is observed in proposed PCA-based S-Box and conventional LUT-based S-Box.

5.1.2 Strict Avalanche Criterion

If one input bit changes in a Boolean function, then half of the output bits should be changed. For a Boolean function, if f_i is to satisfy SAC the following condition should be satisfied, $f(x) \oplus f(x \oplus \alpha)$ should be balanced, where the Hamming weight of α is 1 and SAC is denoted by Υ_S.

$$\mathrm{SAC}_{f_i} = \max_{1 \le i \le n} |2^{n-1} - \sum_{x \in B^n} f(x) \oplus f(x \oplus c_i^n)| \tag{6}$$

In a S-Box, $f : B^n \rightarrow B^m$, and hence, there exists m number of functions $\mu = \{f_1, f_2, \ldots f_m\}$. In the n variable function, B^n consists of all the possible input which is basically 2^n different inputs, and c_i^n consist of all the element in B^n, whose Hamming weight is 1 .

$$\Upsilon_S = \max(\mathrm{SAC}_\mu) \tag{7}$$

If the value of SAC is less for the observed ciphers, then the cipher is more difficult to cryptanalysis. The achieved value of SAC ranges between [0,128], and the best value is observed as 14 for more than 26% of rules, as shown in Fig. 6. The PCA S-Box in terms of SAC have better performance than that of classical S-Box.

5.1.3 Nonlinearity

The nonlinearity of a Boolean function f is the minimum distance of the function to the set of affine functions and represented by $\Re_S$.

$$N_f = \min[d(f, g)], \text{ where } g \in A_n \tag{8}$$

where A_n is the set of all the affine function.

$$d(f, g) = 2^{n-1} - 2^{-1}(\langle \eta, \beta \rangle) \tag{9}$$

where η and β represent the binary sequence of f and g, respectively, and $\langle \eta, \beta \rangle$ define the scalar product of sequence.

$$N_f = 2^{n-1} - 2^{-1}, [\max(\langle \eta, \beta_j \rangle)] \tag{10}$$

where β_j belongs to the sequence of all linear function. In a S-Box, $f : B^n \rightarrow B^m$, and hence, there exists m number of functions $\mu = \{f_1, f_2, \ldots f_m\}$.

$$\Re_S = \min(N_\mu) \tag{11}$$

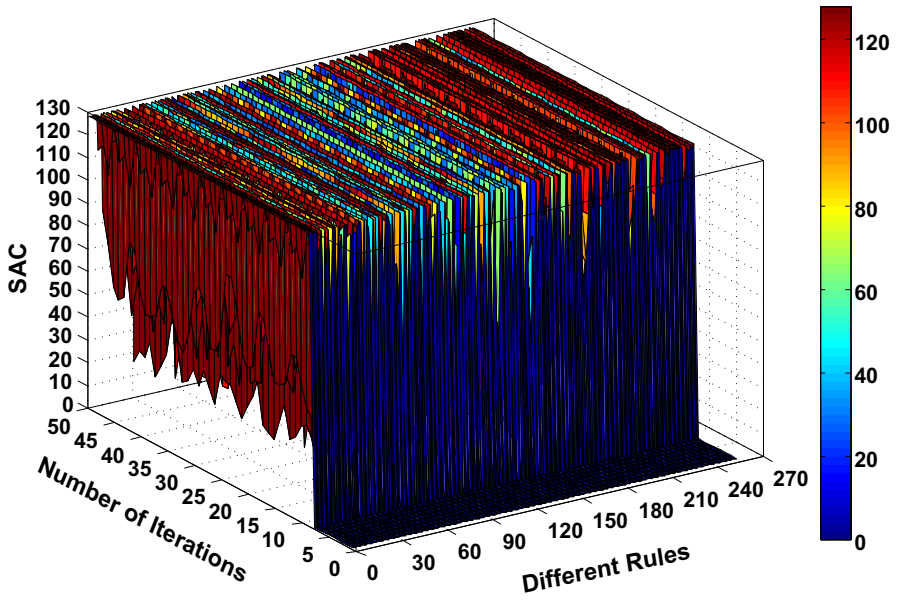

Fig. 6 Value of SAC with different rules

If the achieved value of nonlinearity is significantly high for the observed ciphers, then the cipher is more hard to cryptanalysis. It is observed that the value of nonlinearity varies from [0,109]. Moreover, we found that the achieved value of nonlinearity is more than 100 for 15% of 256 CA rules as shown in Fig. 7, which indicates that the performance of PCA S-Box is comparable to that of classical LUT-based S-Box.

5.1.4 Input/Output Entropy

The entropy of a Boolean function f is the amount of information in input bits, if the output bits are known. There exist 2^n possible inputs and 2^m outputs for a Boolean function of n input and m output. The (i, j)th input/output bit-to-bit entropy of S-Box is computed with $H(\frac{x_i}{f_j(x)})$ and represented by H_S.

$$H = \min[H(\frac{x_i}{f_j(x)})]\ [i \in \{1, n\},\ j \in \{1, m\}] \tag{12}$$

The entropy H_S of the output Boolean function f is given by

$$H_{f_i}(P_i) = P_i \log_2\left(\frac{1}{P_i}\right) + (1 - P_i)\log_2\left(\frac{1}{1 - P_i}\right) \tag{13}$$

where P_i is the fraction of 1s in the output. In a S-Box, $f : B^n \rightarrow B^m$, and hence, there exists m number of functions $\mu = \{f_1, f_2, \ldots f_m\}$.

$$H_S = \min(H_\mu) \tag{14}$$

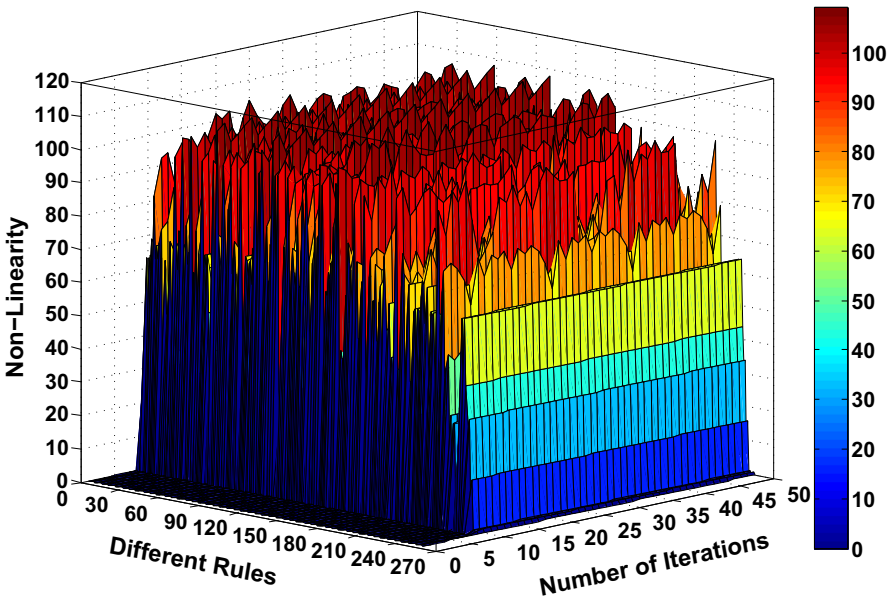

Fig. 7 Nonlinearity with 256 rules

If the entropy value of cipher is high, then the cipher is difficult for cryptanalysis. The entropy value observed for PCA S-Box ranges from [0,1]; the best value attained is 0.99, and the entropy value for most of the CA rules is between [0.95, 0.99], as shown in Fig. 8. The achieved values for conventional LUT S-Box are also depicted in Table 5. The performance of PCA-based S-Box with respect to entropy is better than that of classical S-Box.

5.1.5 Correlation Immunity Bias

A Boolean function f is said to satisfy a correlation immunity bias of order l if it is a statistically independent combination of any l input bits. Mathematically, if l input bits are fixed, then we can get ${}^{n}C_{l}\, 2^{l}\, g$ functions and the correlation immunity bias is denoted by $\Phi_S(l)$.

$$\mathrm{CIB}_f(l) = \max|2^l * W(g_j) - W(f)| \tag{15}$$

where $W(g_j)$ belongs to the Hamming weight of all the possible function keeping l bits in the function f fixed. $W(f)$ corresponds to the Hamming weight of function f. In a S-Box, $f : B^n \rightarrow B^m$, and hence, there exists m number of functions $\mu = \{f_1, f_2, \ldots f_m\}$.

$$\Phi_S(l) = \max(\mathrm{CIB}_\mu) \tag{16}$$

If the value of CIB for cipher is less, then the cipher is more difficult for cryptanalysis. The observed value of CIB for PCA S-Box ranges from [0,128]; the best value achieved is 0, and the values less than 14 are for 23% of 256 CA rules, as shown in Fig. 9. The noticed value of classical S-Box is 14 as indicated in Table 5. From the

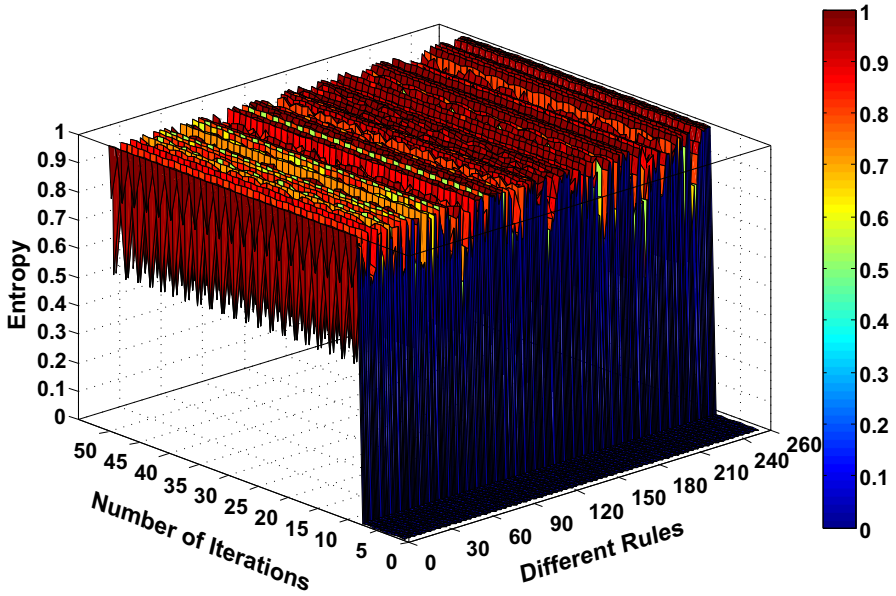

Fig. 8 Entropy with different rules

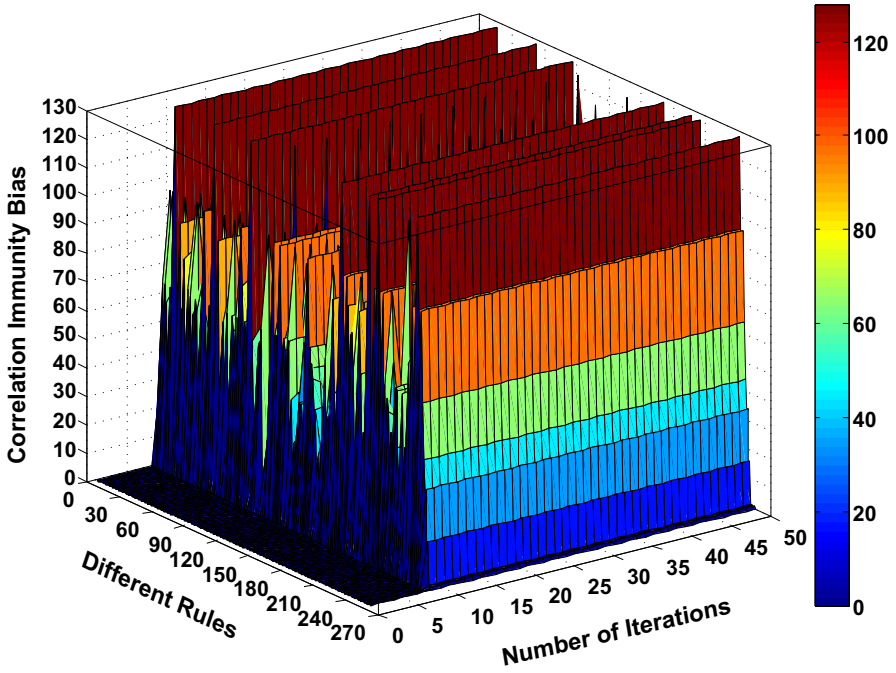

Fig. 9 CIB with different rules

Table 4 CIB, SAC, NL, entropy values for PCA-based S-Box and standard AES S-Box using cryptographic properties

Rule No.	No. of time Steps	Nonlinearity	Entropy	CIB	SAC
30	25	108	0.9823	20	16
30	48	106	0.9972	8	20
45	14	102	0.9956	10	20
45	14	102	0.9956	10	20
57	34	108	0.9857	18	20
57	21	106	0.9914	14	12
75	14	102	0.9956	10	20
86	25	108	0.9823	20	16
86	48	106	0.9972	8	20
89	14	102	0.9956	10	20
99	22	109	0.9857	18	20
135	25	108	0.9823	20	16
135	48	106	0.9972	8	20
149	24	108	0.9925	13	14
149	38	102	0.9972	8	20
CA S-Box [4]	NP	128	NP	NP	128
		0	NP	NP	0
CA S-Box [20]	NP	128	NP	NP	128
		0	NP	NP	0
Hussain et al. [8]	NA	105	NP	NP	16
		96	NP	NP	10
Clark et al. [7]	NA	90	NP	19	44
		100	NP	24	48
Millan et al. [12]	NA	80	NP	NP	16
			NP	NP	18
Nedjah et al. [22]	NA	70	NP	NP	NP
		102	NP	NP	NP
Standard	Polynomial				
AES S-Box	$x^8 + x^4 + x^3 + x + 1$	112	0.9887	16	14

* NA means not applicable
* NP means not provided

above observation, the PCA-based S-Box provides remarkable performance than that of classical S-Box.

The comparative performance of proposed PCA-based S-Box, CA-based S-Box and conventional LUT-based S-Box in terms of security using cryptographic properties is shown in Tables 4 and 5. In Table 4, the nonlinearity value of CA-based S-Box in terms varies from [0,128]; the best value obtained is 128. The achieved value of SAC [0,128] is for CA-based S-Box [4,20]; the observed best value is 0. Table 4 shows that the proposed PCA-based S-Box attained 10% better nonlinearity compared with that

Table 5 Security of LUT-based S-Box using cryptographic properties

Entropy (H)	SAC	CIB	Nonlinearity	AES (Polynomial)
0.9887	16	14	112	$x^8 + x^4 + x^3 + x + 1$.

Table 6 Hardware results of the proposed AES algorithm with PCA-based S-Box

AES	Tech	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kim [27]	0.25 µm	4000	0.02	0.1	870	174
Eslami [25]	0.18 µm	–	7.55	13.56	248	138
Manoj [16]	0.18 µm	–	0.0512	1	500	25.60
Kaps [17]	0.13 µm	4070	0.0238	0.5	534	24.56
Proposed AES algorithm	0.18 µm	4547	3.259	13.69	244	58.702
Proposed AES algorithm	0.13 µm	3971	1.02	13.69	244	18.275

of Clark et al. [7] and Millian et al. [12]. The value of CIB is 15% better than that of Clark et al. [7]. The attained value of nonlinearity and SAC for proposed PCA-based S-Box is comparatively better than Hussain et al. [8].

The PCA-based S-Box is flexible and dynamic in nature, and it is also found that PCA S-Box provides enough level of security compared to LUT-based S-Box.

5.2 Architectural Design

In order to validate the proposed architecture, AES algorithm with PCA-based S-Box is implemented using verilog, verified on FPGA board and synthesized with Cadence RTL compiler. The proposed architecture operates at different clock frequency with TSMC 0.18-µm technology (core voltage of 1.62 V) and UMC 0.13-µm technology (core voltage of 1.08 V) under worst-case conditions. The total time consumed to encrypt 128 bits of plain text is calculated by Latency = Clock cycles × Time period. The performance comparison of AES with PCA-based S-Box and AES with LUT-based S-Box is presented in Table 6 in terms of area, power dissipation, energy consumption and operating frequency. It can be noted that in our proposed PCA-based S-Box realization, the number of iterations was considered as 20 clock cycles to compute the PCA S-Box and the total time taken to encrypt 128 bits of plain text using AES algorithm with PCA-based S-Box is 244 clock cycles (Table 6).

The number of gates utilized for LUT-based S-Box and composite field arithmetic-based S-Box realization was 696 and 294, respectively, with 0.11-µm [5], where in case of the proposed dynamic PCA-based S-Box realization, the number of gates utilized is 113 and 116 using 0.18 and 0.13-µm technology libraries. Sumio et al. [2] presented optimized low-power S-Box architecture for AES which consumes power

of 29 μW at 10 MHz using 130 μm CMOS technology, whereas our proposed PCA-based S-Box at 10 MHz using 130 μm CMOS technology consumes power of 10 μW. It can be easily seen that our proposed PCA-based S-Box consumes 65% less power than the existing work [2]. The work reported in [25] needs power consumption of 7.55 mW for encryption with 0.18-μm technology operated at 13.56 MHz frequency, while our proposed work of AES with PCA-based S-Box operates at 13.69 MHz clock frequency, power consumption of 3.259 mW for encryption and area of 0.184 mm^2 which is 58% less compared to Eslami et al. [25]. The ASIC implementation of AES algorithm with composite field arithmetic-based S-Box using 0.18-μm technology takes 500 clock cycles to complete encryption of 128-bit plain text, when operated at 1 MHz frequency, and the power consumption is 51.20 μW [16]. The power dissipation and energy consumption of AES algorithm with proposed PCA-based S-Box operated at 1 MHz frequency using 0.18-μm technology is 94.07 μW and 22.95 nJ, whereas the energy consumption in Manoj et al. [16] is 25.60 nJ; there is slight decrease in energy consumption by 10% than in the existing work [16]. Our result shows 28% reduction in energy consumption compared to the results of Kaps et al. [17]. It is clear from Table 4 that the proposed PCA-based S-Box outperforms in terms of power dissipation and energy consumption compared with the existing works.

6 Conclusion

In this paper, we proposed a ultralow-power, less area architecture of AES with dynamic PCA-based S-Box for WBAN application. We also evaluated the architecture through simulation and synthesis for ASIC implementation. Unlike the design in [2,6,17,25,27], the proposed design requires few logic elements; hence, there is a reduction in power, energy and chip area compared to conventional AES with LUT-based S-Box. We have achieved comparable performance in terms of security for dynamic PCA-based S-Box with that of classical LUT S-Box using cryptographic properties. The design was synthesized using Cadence RTL compiler to evaluate area, power and frequency of operation. The maximum operating frequency achieved is 536 MHz for TSMC 0.18 μm technology, achieving an area of 0.189 mm^2 and power consumption of 98.33 mW. UMC 0.13 μm technology achieved an area of 0.072 mm^2 and power consumption of 32.059 mW with operating frequency of 769 MHz. Therefore, it has been observed that AES with dynamic PCA-based S-Box is an ultralow-power and low energy consumption encryption algorithm and hence suitable for WBAN applications.

References

1. Advanced Encryption Standard (AES),Federal Information Processing Standards Publication 197 Std., (2001)
2. A. Bechtsoudis, N. Sklavos, Side channel attacks cryptanalysis against block ciphers based on FPGA devices. in *Proceedings of 2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (2010), pp. 460–461
3. J.A. Clark, J.L. Jacob, S. Stepney, The design of S-boxes by simulated annealing. New Gener. Comput. **23**(3), 219–231 (2005)

4. Y. Eslami, A. Sheikholeslami, P.G. Gulak, S. Masui, K. Mukaida, An area-efficient universal cryptography processor for smart cards. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **14**(1), 43–56 (2006)
5. M. Feldhofer, S. Dominikus, J. Wolkerstorfer, Strong authentication for RFID systems using the AES algorithm. in *Cryptographic Hardware and Embedded Systems-CHES 2004, ser*. Lecture Notes in Computer Science Vol. 3156 (Springer, Berlin, Heidelberg, 2004), pp. 357–370
6. B.R. Gangadari, S. Ahamed, R. Mahapatra, R. Sinha, Design of cryptographically secure AES S-box using cellular automata. in *Proceedings of International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015* (2015), pp. 1–6
7. T. Good, M. Benaissa, Very small FPGA application-specific instruction processor for AES. IEEE Trans. Circuits Syst I Regul. Pap. **53**(7), 1477–1486 (2006)
8. A. Hodjat, I. Verbauwhede, Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors. IEEE Trans. Comput. **55**(4), 366–372 (2006)
9. I. Hussain, T. Shah, M.A. Gondal, W.A. Khan, Construction of cryptographically strong 8×8 S-boxes. World Appl. Sci. J. **13**(11), 2389–2395 (2011)
10. IEEE Standard for Local and metropolitan area networks—Part 15.6: Wireless Body Area Networks, Std., (2012)
11. H. Kapoor, G.B. Rao, S. Arshi, G. Trivedi, A security framework for NoC using authenticated encryption and session keys. Circuits Syst. Signal Process. **32**(6), 2605–2622 (2013)
12. J.P. Kaps, B. Sunar, Energy comparison of AES and SHA-1 for ubiquitous computing. in *Emerging Directions in Embedded and Ubiquitous Computing, ser*. Lecture Notes in Computer Science Vol. 4097 (Springer, Berlin, Heidelberg, 2006) pp. 372–381
13. M. Kim, J. Ryou, Y. Choi, S. Jun, Low power AES hardware architecture for radio frequency identification. in *Advances in Information and Computer Security, IWSEC 2006. ser*. Lecture Notes in Computer Science Vol. 4266 (Springer, Berlin, Heidelberg, 2006), pp. 353–363
14. S. Kumar, V.K. Sharma, K.K. Mahapatra, An improved VLSI architecture of S-box for AES encryption. in *Proceedings of 2013 International Conference on Communication Systems and Network Technologies (CSNT)* (2013), pp. 753–756
15. H. Kuo, I. Verbauwhede, Architectural optimization for a 1.82Gbits/sec VLSI implementation of the AES Rijndael algorithm. in *Cryptographic Hardware and Embedded Systems CHES 2001, ser*. Lecture Notes in Computer Science Vol. 2162 (Springer, Berlin, Heidelberg, 2001), pp. 51–64
16. H. Li, Efficient and flexible architecture for AES. IEEE Proc. Circuits Devices Syst. **153**(6), 533–538 (2006)
17. W. Millan, How to improve the nonlinearity of Bijective S-boxes. in *Third Australasian Conference on Information Security and Privacy.ser. ACISP '98*. (Springer-Verlag, London, 1998), pp. 181–192
18. S. Morioka, A. Satoh, An optimized S-box circuit architecture for low power AES design. in *Cryptographic Hardware and Embedded Systems-CHES 2002, ser*. Lecture Notes in Computer Science Vol. 2523 (Springer, Berlin, Heidelberg, 2003), pp. 172–186
19. S. Morioka, A. Satoh, A 10-Gbps full-AES crypto design with a twisted BDD S-Box architecture. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **12**(7), 686–691 (2004)
20. S. Nandi, B.K. Kar, P.P. Chaudhuri, Theory and application of cellular automata in cryptography. IEEE Trans. Comput. **43**(12), 1346–1357 (1994)
21. National Institute of Standards and Technology, FIPS PUB 46-3: Data Encryption Standard (DES), (Oct. 1999), super-sedes FIPS, 46-2
22. N. Nedjah, L.d M. Mourelle, Designing substitution boxes for secure ciphers. Int. J. Innov. Comput. Appl. **1**(1), 86–91 (2007)
23. A. Satoh, S. Morioka, K. Takano, S. Munetoh, A compact Rijndael hardware architecture, with S-Box optimization. in *Advances in Cryptology, ASIACRYPT*. Lecture Notes in Computer Science Vol. 2248 (Springer, Berlin, Heidelberg, 2001), pp. 239–254
24. T.M. Sharma, R. Thilagavathy, Performance analysis of advanced encryption standard for low power and area applications. in *Proceedings of 2013 IEEE Conference on Information Communication Technologies (ICT)* (2013), pp. 967–972
25. M. Szaban, F. Seredynski, CA-based generator of S-boxes for cryptography use. in *Proceedings of 2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)* (2010), pp. 1–8

26. M. Szaban, F. Seredynski, Dynamic cellular automata-based S-Boxes. in *13th International Conference Computer Aided Systems Theory-EUROCAST 2011. ser*. Lecture Notes in Computer Science Vol. 6927 (Springer, Berlin, Heidelberg, 2012) pp. 184–191
27. X. Zhang, K.K. Parhi, High-speed VLSI architectures for the AES algorithm. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **12**(9), 957–967 (2004)
28. X. Zhang, K.K. Parhi, On the optimum constructions of composite field for the AES algorithm. IEEE Trans. Circuits Syst. II Express Briefs **53**(10), 1153–1157 (2006)