Wireless Pers Commun (2015) 85:305–323
DOI 10.1007/s11277-015-2739-x

PSP CO_2: An Efficient Hardware Architecture for AES Algorithm for High Throughput

P. Karthigaikumar[1] · N. Anitha Christy[2] · N. M. Siva Mangai[2]

Published online: 17 May 2015

Abstract In this modern era, communication plays an important role in a human's life. Also information security is a significant aspect of all types of communication. Now a day all the communications are carried out in wireless medium. It is necessary to transmit the confidential data in wireless media in a secure manner. Cryptography is a technique to protect the electronic data in a communication network. Efficient hardware architecture to implement the Advanced Encryption Standard (AES) algorithm for high throughput and less area is presented in this paper. In the proposed architecture the throughput is increased by using the Parallel Sub-Pipeline (PSP) architecture for the AES algorithm, the techniques like composite field arithmetic (CFA), on the fly key expansion and order change are combined in order to reduce the area. Also different combination like PSP plus on the fly, PSP plus CFA and PSP plus order change are explored in this research. Based on synthesis report and the throughput, it is suggested that the proposed PSP plus CFA plus On the fly plus Order change (PSP CO_2) produces reasonably high throughput and less area compared to other combination. The proposed PSP CO_2 architecture is implemented in field programmable gate array. This implementation achieves a throughput of 52.29 Gbps at a frequency of 450.045 MHz on Xilinx Virtex XC6VLX75T device which is reported to be higher than all the other implementations in the literature survey.

Keywords Cryptography · AES · Parallel Sub-Pipeline · FPGA · Throughput · Network security

✉ P. Karthigaikumar
p.karthigaikumar@gmail.com

[1] Department of Electronics and Telecommunication Engineering, Karpagam College of Engineering, Coimbatore, India

[2] Department of Electronics and Communication Engineering, Karunya University, Coimbatore, India

1 Introduction

Network security yields a safe and secure communication. It consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network accessible resources. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs involving transactions and communications among businesses, government agencies and individuals.

Cryptography provides way for securing electronic information. Many cryptographic algorithms were invented in the past few years. In that only few algorithms are withstanding against the third party. There was a question of long term security of the electronic information. In 2001, the National Institute of Standards and Technology (NIST) [1] tested few algorithms and announced that the Advanced Encryption Standard (AES) algorithm is the best cryptographic algorithm. Hence, in this paper the AES algorithm has been chosen and the architecture is modified in order to obtain high throughput.

Also in this paper, several area optimisation techniques like composite field arithmetic (CFA), order changing and on the fly key expansion have been implemented to reduce the area of the AES algorithm. Because of high throughput and less area, the proposed architecture is used for the real time image processing application, i.e., image encryption/decryption.

1.1 Motivation

It is mandatory to protect the confidential data over the internet. The Software implementation has the advantage of upgradability, easy usage, portability and flexibility. Since the hardware implementation provides a high physical security [2] than the software implementation, the proposed AES architecture is implemented in FPGA. The FPGA combines the flexibility and ease of upgrade that fulfils the advantage of software implementation with much improved physical security. The motivation of this paper is to present the hardware architecture for AES algorithm with high throughput and less area, so that it can be applied for the low end embedded applications.

This paper is organized as follows. Section 2 gives the relevant works of various authors reported in the literature. Section 3 reviews the basic AES encryption/decryption algorithm. Section 4 gives the contribution of this paper. Section 5 briefly describes the proposed methodology for increasing the throughput and for reducing the area. The implementation results and the performance comparison with the existing literature are discussed in the Sect. 6. The conclusion and the future work are stated in the last section.

2 AES Algorithm

Information security i.e. a secure data transmission can be obtained by employing effective and reliable encryption algorithms. There are several encryption algorithms which serves this purpose. AES [3] is one of the strongest cryptographic algorithms. It is approved by the NIST. AES is a symmetric block cipher which can encrypt and decrypt the information. AES takes input in blocks of 128, 192 or 256 bits and uses keys of lengths 128, 192 and 256 bits. The 128 bits are divided into 16 bytes and are put in a 4×4 array. A Byte is the

basic unit for processing in AES algorithm. A Byte can be represented as a polynomial. The encryption and decryption processes of AES algorithm is shown Fig. 1.

2.1 Steps of AES Algorithm

Encryption and decryption processes [4–7] consist of several steps. They are AddRoundKey, ShiftRows/InverseShifRows, ByteSubstitution step/Inverse ByteSubstitution, MixColumns/Inverse MixColumns step. These steps constitute a round. The number of rounds depends upon the size of the key. For the key sizes 128, 192, 256 bits, the corresponding number of rounds is 10, 12 and 14. Next sub-section provides brief explanation about each step.

2.1.1 AddRoundKey

In this step, the input data and the initial key are Xored. This is the first step in encryption process. The Key Expansion unit expands the initial key so that it can be used in the further AddRoundKey steps.

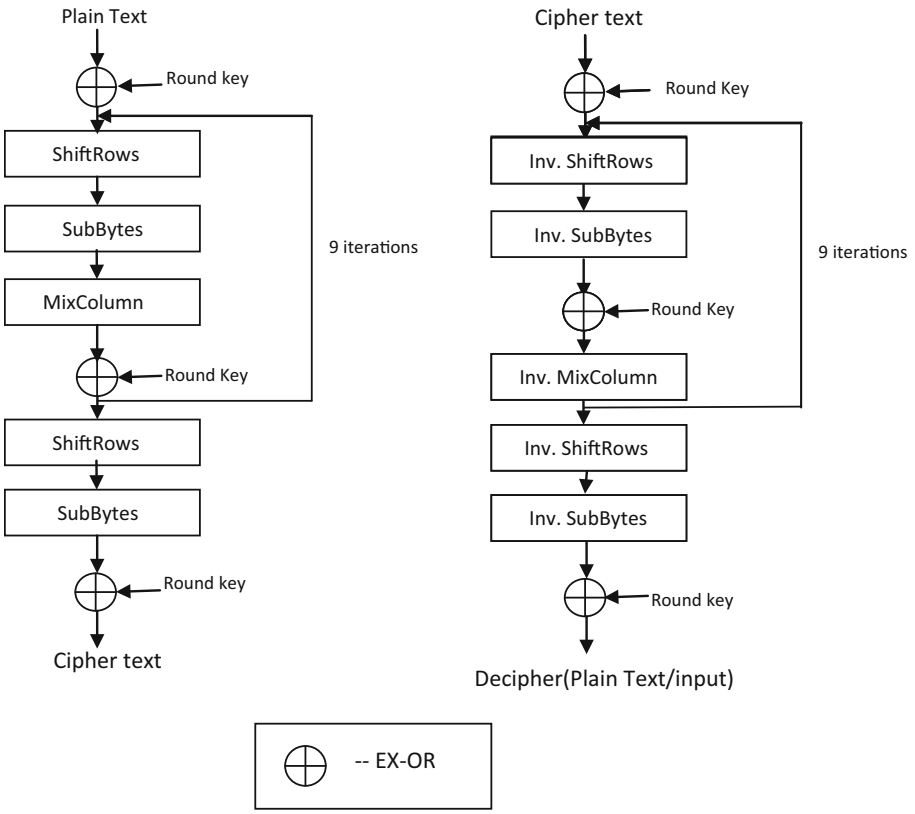


Fig. 1 Encryption and decryption process of AES

2.1.2 ByteSubstitution/Inverfse ByteSubstituion

The ByteSubstitution in encryption and the Inverse ByteSubstitution in decryption are the only non-linear transformations in the rounds of AES algorithm. Here each Byte of the state array, which is considered as an element of GF (2^8), is transformed to another byte by referring a Look-Up-Table (LUT).

2.1.3 ShiftRows/Inverse ShiftRows

In the ShiftRows and Inverse ShiftRows step, the rows of the state array are shifted cyclically to the left and right respectively. The first row remains unchanged. The second, third and fourth rows are shifted by one, two and three offsets to the left for the ShiftRows transformation. For the Inverse ShiftRows step, the second, third and fourth rows are shifted to the right by one, two and three offsets.

2.1.4 MixColumn/Inverse MixColumn

In these steps, each column of the state array is multiplied with a polynomial. The MixColumn transformation can be expressed in the matrix form as follows,

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

The Inverse MixColumn step is expressed as follows,

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix}$$

3 Related Research

The implementation of the cryptographic algorithm in hardware has been carried out since past few years. Several architectures have been proposed to implement the AES algorithm by different authors.

Samiee et al. [8] designed a fully sub-pipelined encryptor with six substages in each round unit and achieved a throughput of 43.71 Gbps on a Xilinx XC2VP207fg676 device. The author also presented a efficient key expansion architecture suitable for six sub-pipelined architecture.

Reddy et al. [9] proposed two architectures—Pipeline and Sub-Pipeline. The architectures were coded using Verilog hardware description language (HDL) and simulated using Cadence ncsim. In this paper, pipelined AES is implemented using LUT. This design is implemented on XC5VLX110T-1 device. The area of 4611 slices and the throughput of 13.238 Gbps is achieved in this method. The other design uses Sub-Pipeline architecture and it occupies an area of 8896 slices and the throughput of 25.89 Gbps.

Zhang et al. [10] presented novel high speed architectures for the hardware implementation of the AES algorithm. CFA is employed to reduce the area and different implementations for the inversion in subfield GF (2^4) are compared in this paper. A fully Sub Pipelined encryptor with seven sub stages in each round is implemented and achieved a throughput of 21.56 Gbps on Xilinx XCV1000 device.

Kamal et al. [11] implemented area optimization for the AES algorithm and it required 2732 slices of a Xilinx Virtex-II XC2V1000bg575 device, runs at a maximum clock speed of 98.95 MHz and achieves a throughput of 29.32 Mbps.

Hammad et al. [12] designed a new efficient architecture for high speed AES encryptor using CFA in Byte Substitution Round. The architecture is presented with a multistage sub pipelined architecture that allows a high throughput. This design achieves a throughput of 39.053 Gbps for nine pipelining stages with an operational frequency of 305.1 MHz. This design also achieves a throughput of 27.94 Gbps on Xilinx Virtex 2 device with an operational frequency of 218.3 MHz.

Fan and Hwang [13] proposed a sequential and fully pipelined AES realization using Xilinx ISE 7.1 synthesizer and uses an efficient low cost AddRoundKey architecture for real time key generations. The sequential AES design achieves a throughput of 0.876 Gbps with an operational frequency of 75.3 MHz. The pipelined AES design achieves a throughput of 28.4 Gbps with an operational frequency of 222 MHz.

Swankoski et al. [14] proposed a parallel architecture in which internal hardware functionality is not duplicated but reused. This created a reasonably compact single block, which is ideal for duplication. This allowed multiple users to share the same hardware, as spatial isolation is achieved by the physical separation of individual encryption blocks. This algorithm achieved a throughput of 18.80 Gbps and an area of 23,979 slices in Virtex 2 pro FPGA device with ten parallel blocks.

Yoo et al. [15] presented a high speed parallel pipelined architecture is proposed in order to get high throughput. The AES block cipher is implemented with Virtex II Pro FPGA using 0.13 μm and 90 nm technology. By using an efficient inter-round and intra-round pipeline design, it achieves a high throughput of 29.77 Gbps in encryption.

Jose et al. [16] implemented an AES-128 algorithm using parallelism, pipelining, partial and dynamic reconfiguration. Using partial and dynamic reconfiguration, the subkeys which are contained in LUT's are modified in this paper. Three hardware languages Handel-C, VHDL and JBits are combined with partial and dynamic reconfiguration, pipelining and sub-pipelining. It reaches a throughput of 24.922 Gbps on Xilinx XC2V6000-6.

Jyrwa et al. [17] presented a hardware implementation of AES algorithm. In this paper, the authors have worked with an iterative architecture and modifications such as merging of SubBytes and ShiftRows, LUTs for decryption have been successfully implemented. The encryption and decryption process of Rijndael algorithm was done in VHDL language and the corresponding FPGA implementation in the device XC2VP30 yields an area of 6211 slices and a throughput of 1.458 Gbps is obtained.

4 Contribution of This Paper

From the related work, it is observed that there are number of architecture like parallel, pipeline, sub-pipeline, parallel pipeline, sub-pipeline CFA are used to increase the throughput of AES. Here in this paper, parallel sub-line architecture is used to increase the

throughput, CFA based S-box, on the fly key expansion and order change techniques are combined to reduce the area which is not reported in the earlier work.

The proposed AES architecture was synthesized in a Xilinx ISE12.2i tool and simulated using ISIM. This prototyped AES in FPGA achieves a high throughput of 52.29 Gbps and occupies an area of 1094 slices.

5 Proposed Methodology

5.1 Parallel Sub-Pipeline Architecture for High Throughput

This architecture combines the parallel and sub-pipeline architecture. The Sub-Pipeline architecture has the advantage of processing more number of inputs. This is done by storing the intermediate results in the register. The area is increased due to increased use of registers, but the throughput is increased to a greater extent. The parallel architecture has the advantage of processing the data in high speed by dividing the total module into small modules and processing it parallely. The parallel architecture [18] also provides high throughput but occupies more area.

The advantages of both the architectures have been merged by combining the Parallel and Sub-Pipeline architecture which is shown in the following Fig. 2.

In the proposed technique, the input 128 bits is divided into four 32 bits and is given to separate hardware components, thus achieving the parallel architecture. Registers are inserted in intra round and inter round. The inner round consists of AddRoundkey, SubBytes, ShiftRows and MixColumn. These comprise one round. In AES-128, there are ten rounds. The intra round consists of these ten rounds i.e., from round 1 to round 10. This achieves the Sub-Pipelined architecture [19].

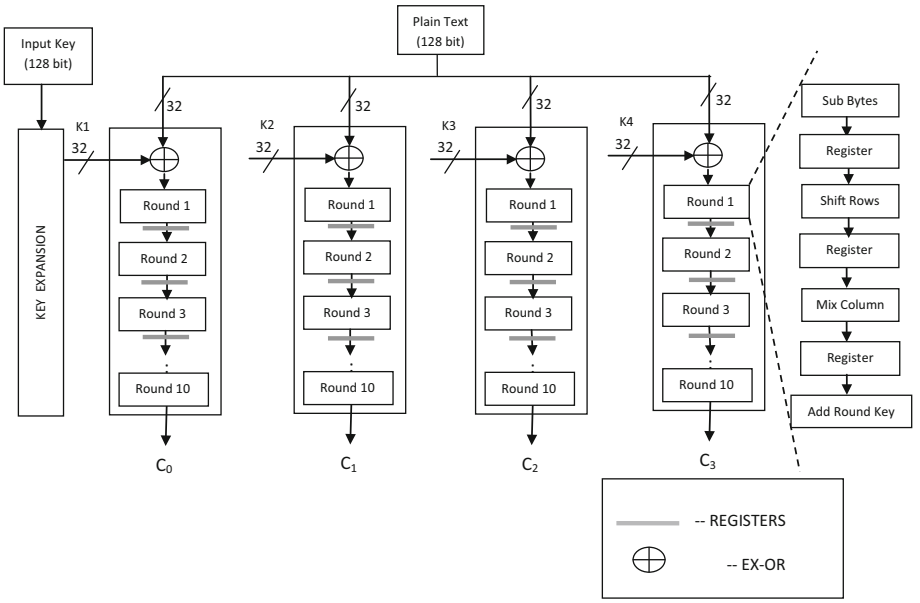


Fig. 2 Parallel Sub-Pipeline architecture

Initially the key is given to the Key Scheduling unit, where the key is expanded by rotating and exor-ing with the Round Constant values. The Key generated from the first round is sent to the next round which is shown in the Fig. 3.

First of all, there will be a exor operation between the input and key and resulting result will be further processed through the rounds.

When the Clock is given, all the four 32 bits input will enter into the round 1. For the next clock, the result from round 1 is stored in the register and is sent to the round 2. At the same time, the next new set of inputs will enter into the hardware. Thus a greater throughput can be achieved from this architecture. This type of architecture increases the area. Inorder to decrease the area, some of the area optimisation techniques are explored and implemented. The different area optimisation techniques alter some of the internal architecture of the AES algorithm. The area optimisation techniques [20] are using CFA in the Sub-Bytes round instead of using LUT for S-Box, by generating the keys on the fly and by changing the order of the AES algorithm is discussed in the next sub-section.

5.2 Techniques to Reduce the Chip Area

5.2.1 Composite Field Arithmetic Structure in S-Box

In LUT based approaches, each Byte in the state array is transformed into another byte by looking at the LUT. This LUT [21] is referred nearly 200 times throughout the AES algorithm. Hence this occupies a large area. In order to reduce the chip area, non-LUT based approaches are preferred.

The SubBytes and the Inverse SubBytes transformations are the most resource consuming operations in the steps of AES algorithm. An area optimized AES algorithm is a high boon to many resource limited applications like RF Identification (RFID) tags, embedded systems applications like Mobile phones and tiny sensor networks. Hence CFA is employed in the SubBytes and Inverse SubBytes steps to reduce the area.

Subbytes Using CFA: Non-LUT based approaches for the SubBytes and its Inverse is implemented using combinational circuits. This circuit calculates the values of the transformed byte on the fly without having the values pre-stored using tables. The implementation of a circuit to find the multiplicative inverse in the GF (2^8) is very difficult and costly. Therefore, CFA [22–26] is employed. By using CFA, an element in the higher order field GF (2^8) is mapped to an element in the lower order field GF (2^4) and can be further mapped to an elementary field GF [$(2^2)^2$]. Now computations can be made in the lower order field. The Multiplicative Inverse can be found in the lower order field using the following polynomial [23],

$$(bx + c)^{-1} = b\left(b^2\lambda + c(b + c)\right)^{-1}x + (c + b)\left(b^2\lambda + c(b + c)\right)^{-1} \quad (1)$$

The combinational circuit for the Multiplicative Inverse is derived using the Eq. (1) and is shown in Fig. 4 [23]. The detailed circuit of the multiplicative inverse is given in [23].

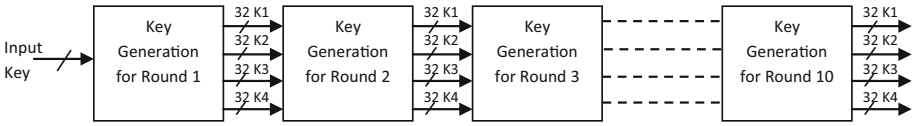


Fig. 3 Key generation unit

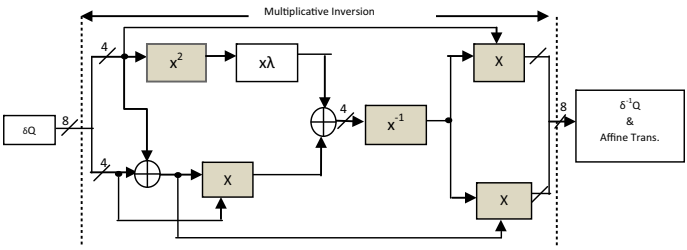


Fig. 4 Byte substitution using CFA

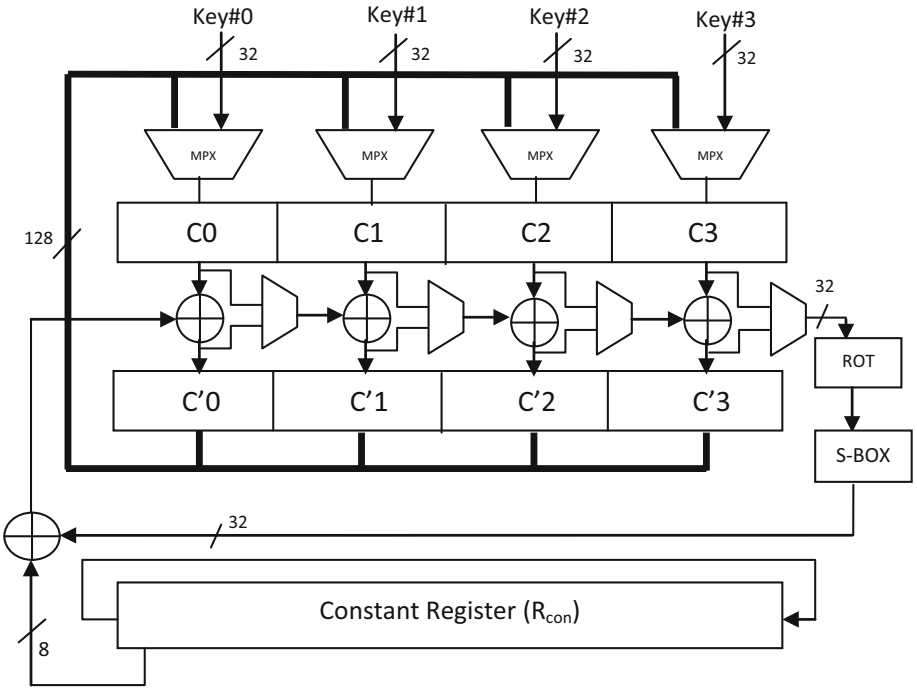


Fig. 5 On the fly key expansion module

Inverse Subbytes Using CFA: The Inverse S-Box can also be built using CFA. The steps involved are

1. Inverse Affine Transformation and Isomorphic mapping
2. Multiplicative Inverse
3. Inverse Isomorphic Mapping

Inverse Affine transformation in matrix form is shown in the Eq. (2)

$$A^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

The Isomorphic Mapping, Multiplicative Inverse, Inverse Isomorphic Mapping steps are the same as the SubBytes step using CFA.

5.2.2 On the Fly Key Expansion Technique

As the name itself suggests, the keys are generated on the fly [27] using this technique. In the LUT based key expansion technique, the pre calculated keys are stored in a particular location. This consumes some amount of area. In order to reduce the area, the keys are computed on the fly and the circuit to generated the on the fly key is shown in the Fig. 5.

The 128 bit input key is divided into 4 bytes (words) as Key #0, Key #1, Key #2, Key #3. The R_{con} register consists of the round constant values to be used for each round. Initially, the Key #3 is rotated and the bytes are substituted from the S-Box. This value is ex-ored with the Round Constant values and is ex-ored with the key #0.

The resulting word and the next key #1 is ex-ored and the resulting word will be the new generated key #2 which will be sent as the input for the next round input. Similarly the operation follows for the consecutive words. Again, the new generated key is fed as input

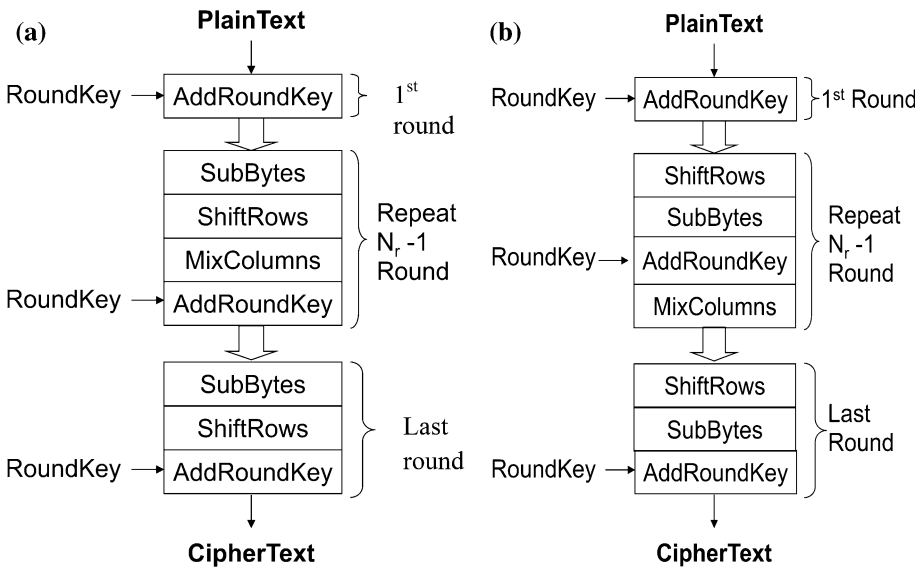


Fig. 6 AES encryption **a** ordinary structure, **b** equivalent structure

to the Key Generation unit for the next set of key for the next round. Thus the process of generating key continues until it reaches the round 10 for AES-128 bit algorithm.

In this paper, a process of generating keys on the fly reduces area but consumes time. This can be overcome by combining this concept with the Parallel Sub-Pipeline architecture, as it saves time by doing the operation in high speed.

5.2.3 Order Changing

In AES design flow, the SubBytes and ShiftRows transformations are commute; that is, a SubBytes transformation immediately followed by a ShiftRows transformation is equivalent to a ShiftRows transformation immediately followed by a SubBytes transformation [28]. The order of the AddRoundKey and MixColumns transformations can also be reversed, provided that the columns (words) of the decryption key schedule are modified using the MixColumns transformation.

Thus the encryption structure in Fig. 6a can be modified to the equivalent structure in Fig. 6b. In this figure, note the change in the sequence of SubByte and ShiftRow and the sequence of AddRoundkey and MixColumn.

Figure 7 shows the straightforward decryption structure of the AES algorithm. One should note that,

- InvShiftRows transformation immediately followed by InvSubBytes transformation is equivalent to InvSubBytes transformation immediately followed by InvShiftRows transformation [29].
- InvMixColumns transformation is linear and hence

$$\text{InvMixCol}(\text{state} \oplus \text{roundkey}) = \text{InvMixCol}(\text{state}) \oplus \text{InvMixCol}(\text{roundkey})$$

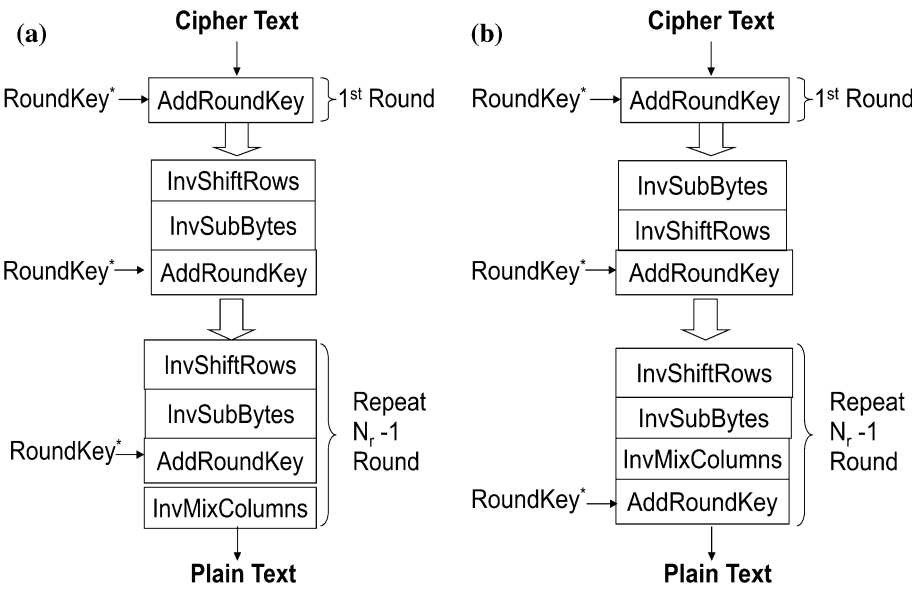


Fig. 7 AES decryption **a** ordinary structure, **b** equivalent structure

Thus the decryption structure in Fig. 7a can be modified to the equivalent structure in Fig. 7b. In this Figure, note the change in the sequence of InvSubByte and InvShiftrow also the change of Mix round column and Add round key.

This order changing approach optimizes the architecture and hence the area is reduced in this technique.

6 Performance Comparison

6.1 Evaluation Metrices

The throughput, area consumption and efficiency are the most important parameters in evaluating the performance of the implementation. The proposed PSP CO_2 architecture is prototyped in Xilinx Virtex XC6VLX75T device. The architecture is simulated on ISIM tool after post place and route. The entire design is synchronized with clock signals. The simulation is done based on the NIST standard inputs and checked for encrypted outputs. The different input parameters given for our algorithm are given in Fig. 8. Mode is set to '1' for encryption and '0' for decryption. Also, the PSP CO_2 architecture is also tested for the image encryption/decryption by linking the MATLAB tool with Xilinx tool.

6.2 RTL Schematic

The RTL schematic of the proposed PSP CO_2 architecture is shown in the Fig. 8. The entity name is specified at the top as encdec32. In this paper, Parallel Sub-Pipeline architecture is proposed for high throughput. In this architecture the 128 bit is divided into four bytes as shown in the figure as input 1, input 2, input 3, input 4. The encrypted and decrypted bits are shown as output from the block.

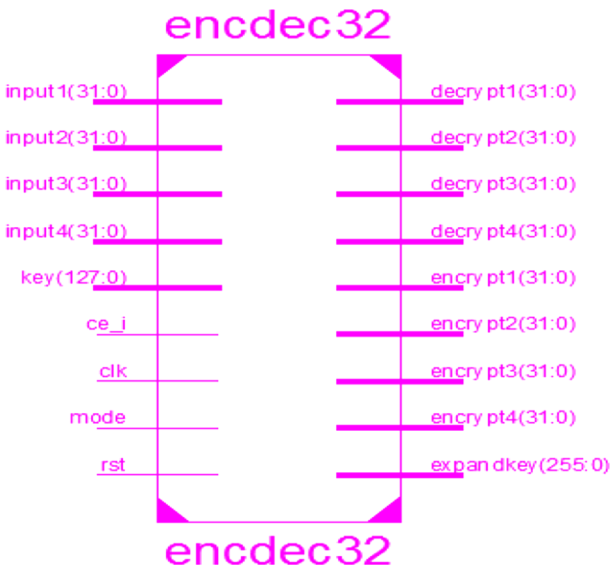


Fig. 8 RTL schematic of the proposed PSP CO_2 AES architecture

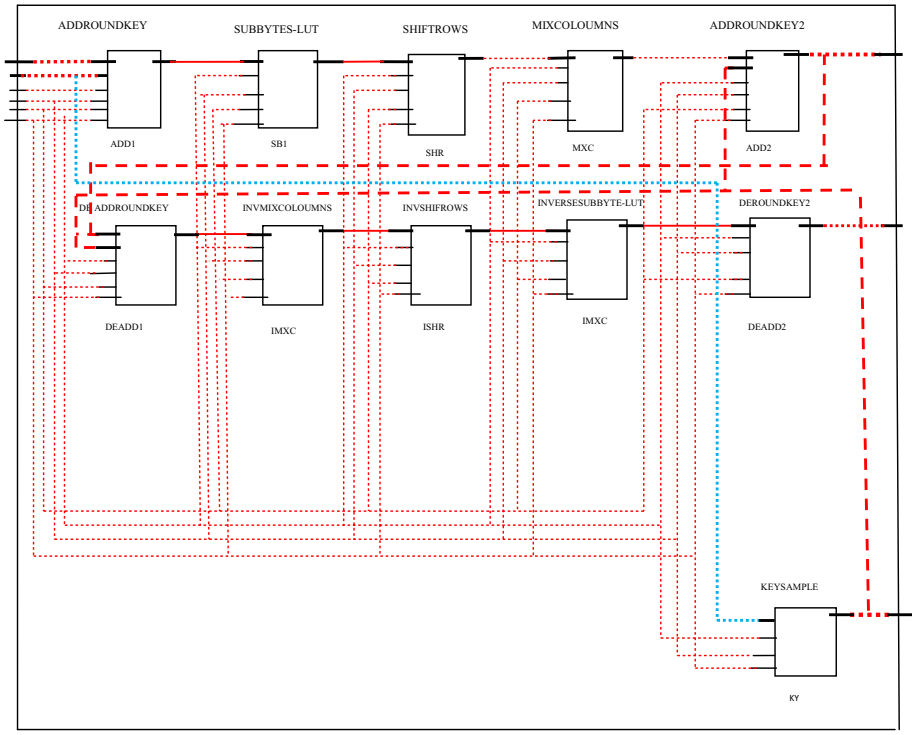


Fig. 9 Detailed internal RTL schematic of the proposed architecture with LUT in the Sub-Bytes round

The inner RTL Schematic of the PSP CO_2 architecture is shown in the Fig. 9. The Fig. 9 shows the different stages of AES algorithm and the inputs and outputs of the consecutive stages.

6.3 Throughput & Area Analysis in FPGA

The design is coded in Very High Speed Integrated Circuit Hardware Description Language (VHDL). Synthesis is done by Xilinx ISE12.2i and the timing simulation is performed by ISIM simulator to verify the functionality of the circuit. The device Virtex 6 XC6VLX75T is the targeted device for FPGA implementation.

Throughput [29] is the speed at which the data is encrypted/decrypted. The throughput is very important in a communication process and this determines the performance of the algorithm.

The throughput is determined by the Eq. (3)

$$\text{Throughput} = \frac{128 \times \text{Number of blocks/cycle}}{\text{Clock Period}} \tag{3}$$

128—indicate the block size of the input data.

The efficiency [30] is calculated by using Eq. (4)

$$\text{Efficiency} = \frac{\text{Throughput}}{\text{Area}} \tag{4}$$

The performance comparison of the proposed PSP CO_2 architecture with the existing architecture reported in literature is shown in the Table 1. Efficiency is the throughput per area which is denoted as Mbps/slices. The throughput and efficiency are calculated for all the architectures in order to estimate and compare the speed of the algorithm.

Granado et al. [16] proposed a methodology to implement the AES algorithm using pipelined and parallel architecture using partial and dynamic reconfiguration. This achieves a throughput of 24.922 Gbps and an area of 3576 slices. The proposed combined architecture achieves a high throughput of 27.68 Gbps in the same device and the area is also higher than the [11], because more number of registers is used in the PSP CO_2 architecture.

Thangkhome et al. [31] proposed two architectures Loop Unrolled and Pipelined architectures and showed that the Pipelined architecture achieved a higher throughput than the Loop Unrolled architecture. The Table 1 shows that the proposed architecture achieves higher throughput than the Pipelined architecture since the higher speed will be achieved through the parallel computing and Pipelined techniques. The proposed architecture achieves a throughput of 28.98 Gbps and an area of 4230 slices. Higher efficiency of 6.85 Mbps is achieved in the proposed architecture.

Table 1 Performance comparison of FPGA Results of the PSP CO_2 with the existing works

S. no	Device	Author	Architecture	Throughput (Gbps)	Area (Slices)	Efficiency (Mbps/slice)
1	XC2V6000-6	[16]	Parallel Pipeline	24.92	3576	6.97
		This work	Proposed	27.68	4484	6.17
2	XC2VP7X	[31]	Loop Unrolled	3.85	2599	1.48
			Pipeline	6.16	3119	1.97
		This work	Proposed	28.98	4230	6.85
3	XC2VP70	[32]	Pipeline	34.7	2389	14.5
		This work	Proposed	35.38	4110	8.60
4	XC4VLX40	[33]	Iterative	0.497	1725	2.88
		This work	Proposed	37.23	5050	7.37
5	XC5VLX75T	[9]	Pipeline	13.238	4611	1.077
			Sub-Pipeline	25.89	8896	2.91
		This work	Proposed	41.88	8978	4.66
6	XC2VP20	[15]	Parallel Pipeline	28.44	6541	4.34
		This work	Proposed	29.23	6925	4.22
7	XC6VLX75T	This work	Iterative	39.07	1953	20
			Pipeline	43.051	1962	21.94
			Sub-Pipeline	48.531	1988	24.41
			Parallel	53.117	2121	25.04
			Parallel Pipeline	57.605	2322	24.80
			Parallel Sub-Pipeline (PSP)	59.59	2597	22.94
			PSP + on the fly	56.35	1893	29.76
			PSP + CFA	54.23	1380	39.29
			PSP + order change	57.45	2125	27.03
			PSP + on the fly + CFA + order change (proposed-PSP CO_2)	52.29	1094	47.79

Zhang and Wang [32] proposed a pipelined architecture for AES algorithm and a throughput of 34.7 Gbps and an area of 2389 is occupied. The PSP CO_2 gives a throughput of 35.38 Gbps and area of 4110 slices. The increase in area is due to the insertion of registers in inter round and intra round.

Iyer et al. [33] proposes an iterative architecture which achieved a throughput of 0.497 Gbps and an area of 1725 slices in Xilinx Virtex 4 device. The PSP CO_2 architecture achieved throughput of 37.23 Gbps and an area of 5050 slices in the same device.

Yoo et al. [15] proposes a parallel pipelined architecture of 28.44 Gbps and an area of 6541 slices in the Xilinx Virtex 2P device. The proposed architecture achieves a throughput of 29.23 Gbps and an area of 6925 slices in the same device.

The throughput of the proposed PSP CO_2 architecture and the throughput of various architectures as proposed by several authors is compared, keeping the device same and is shown in Fig. 10. For example the throughput calculated for the device X2V6000-6 in [16] is compared with the throughput obtained for the same device by applying proposed architecture which is found to be better.

From the Fig. 10, it is observed that the proposed architecture achieves high throughput than the existing works with the same device.

Also, HDL code is written for the existing architectures like Iterative, Pipeline, Sub-Pipeline, Parallel and Parallel Pipeline and synthesized targeting XC6VLX75T device. The throughput and the efficiency of the proposed PSP CO_2 architecture and the other existing architecture are shown in Figs. 11 and 12.

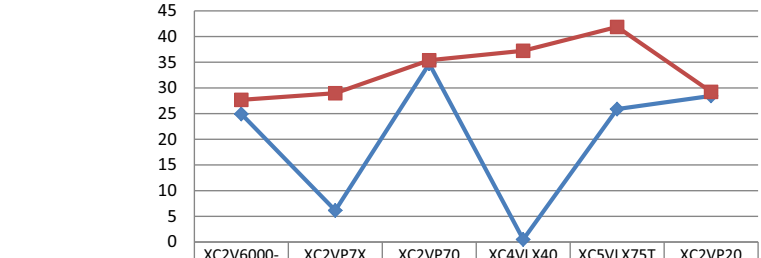
From Fig. 12, it is noted that the proposed PSP plus CFA plus on the fly key expansion plus order change architecture provides higher efficiency than the other architecture.

Hence this paper presents different architectures for different throughput and reduced area, so that the user can chose the architecture based on requirement and application.

6.4 Cryptanalysis of AES Algorithm

There are various possible attacks are reported in [29]. According to [29], potential cryptanalysis performed on AES was not fruitful in retrieving the key or the plain text. The authors in [30] prove that differential cryptanalysis and linear cryptanalysis performed on AES will not able to break and proves to be more secure.

In algebraic attacks which works such that the system is expressed as a multivariate polynomial equations which can be solved to find the key [31]. The eXtended Linearization



	XC2V6000-6 [16]	XC2VP7X [31]	XC2VP70 [32]	XC4VLX40 [33]	XC5VLX75T [9]	XC2VP20 [15]
Existing Architecture	24.92	6.16	34.7	0.497	25.89	28.44
PSP CO2	27.68	28.98	35.38	37.23	41.88	29.23

Fig. 10 Throughput comparison of proposed work with literature

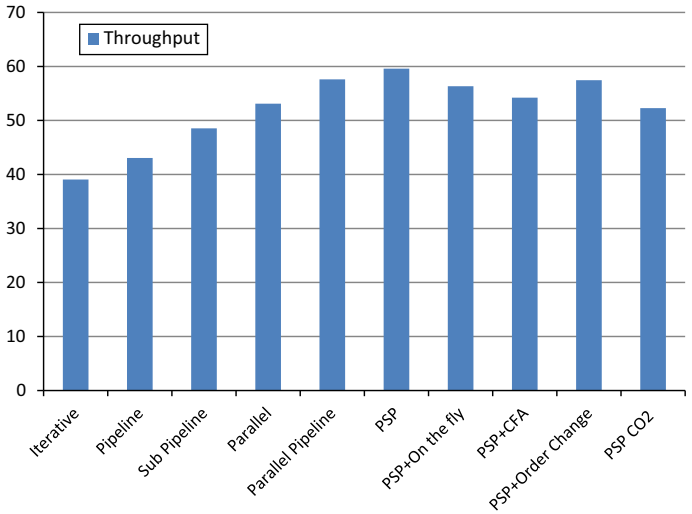


Fig. 11 Throughput comparison of proposed architecture with existing architecture

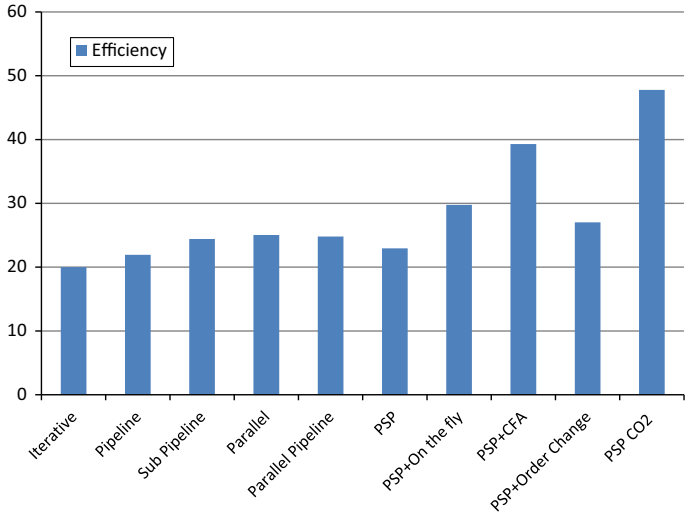


Fig. 12 Efficiency comparison of proposed architecture with existing architecture

(XL) algorithm [32] and the eXtended Sparse Linearization (XSL) algorithm [33] were aimed at solving the systems of equations obtained through crypt analysis. But the number of equations with thousands of unknown variables makes these less feasible for computing the key.

In side channel attacks the variations in observable parameters are noted and crypt-analysis is done on these parameters. Only timing attacks are non invasive, all other side channel attacks like power analysis attack, fault injection attack, electromagnetic radiation are invasive. The probability of these attacks are normally less because of the requirement

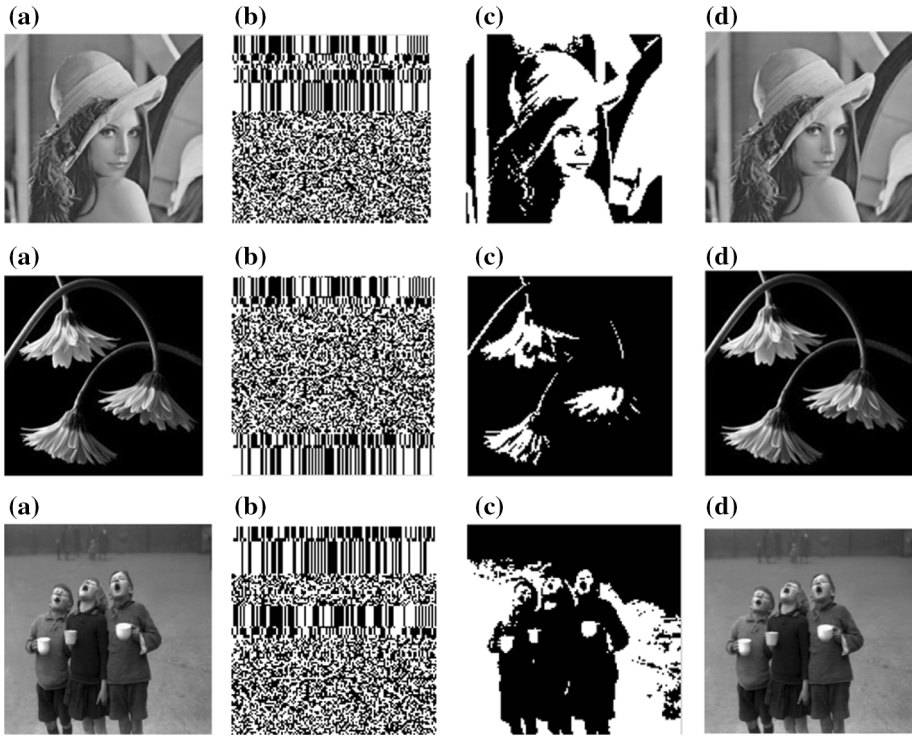


Fig. 13 Image encryption/decryption using the proposed PSP CO_2 architecture of AES. **a** Original image, **b** encrypted image, **c** decrypted binary image, **d** reconstructed image

of precise measuring equipments and the requirement of encrypting device itself. Also there are many countermeasures available to overcome these side channel attacks, like increasing latency, masking of data, shuffling of data after every access etc. Though related key attack [33] was able to crack few rounds of AES 192 and AES 256, a complete attack to break the AES 128 and retrieve its key has not been possible as of now.

These results don't provide a promising cryptanalysis to break the AES-128 algorithm hence we believe that AES-128 algorithm is quiet promising indeed.

6.5 Real Time Image Encryption/Decryption

The proposed PSP CO_2 architecture is also tested for different set of input images. In this research, the gray scale images of different sizes are taken and are resized to 128 × 128 size. The gray scale images are converted to binary images using the command dec2bin in MATLAB. Now, the pixel values of the binary images are 1's and 0's which are then sent to the proposed architecture of AES encoder. In the AES encoder, the plain text/input bits are converted to the cipher/encrypted data. The encrypted data is again sent to the MATLAB using the FILE operation. The encrypted image's pixel values are then sent to the AES decoder. In the AES decoder, the encrypted data is converted to the decrypted data/plain text which is again linked to the MATLAB through the FILE operation in VHDL. The decrypted image obtained will be the binary image which is again converted to

gray scale image by using the command bin2dec. The resulting will be the reconstructed image which will be same as the Original image. Images of Lena, Flow and Frnd are encrypted and decrypted using proposed architecture and are shown in the Fig. 13.

7 Conclusion

This paper presents the FPGA implementation of high throughput and reduced area architecture for AES Encryption and Decryption algorithm for wireless communication. In this research, the Parallel Sub-pipelined architecture yielded a high throughput and a better efficiency for AES Algorithm and the area of the algorithm is optimised using area optimisation techniques such as using CFA in the place of LUT in SubBytes round, On the fly key Expansion technique and by changing the order of the steps in the AES algorithm.

The FPGA implementation of the proposed architecture gives high throughput of 52.29 Gbps and an area of 1094 slices. The image is also tested for its encryption/decryption using the proposed architecture. Thus, the proposed architecture achieves a high throughput and reduced area than all the existing architectures and can be used in real time applications for high security.

References

1. National Institute of Standards and Technology (NIST) (2001). Federal information processing standard publication 197, the Advanced Encryption Standard (AES).
2. Karthigaikumar, P., & Baskaran, K. (2010). An ASIC implementation of low power and high throughput blowfish crypto algorithm. *Microelectronics Journal, 41*, 347–355.
3. Rouvroy, G., Standaert, F.-X., Quisquater, J.-J., & Legat, J.-D. (2004). Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications. *International Conference on Information Technology: Coding and Computing, 2*, 583–587.
4. Harrison, O., & Waldron, J. (2007). AES encryption implementation and analysis on commodity graphics processing units. In *9th workshop on cryptographic hardware and embedded systems* (*CHES 2007*), (Vol. 4727, pp. 209–226).
5. Alma'aitah, A., & Abid, Z. E. (2010). Area efficient high throughput sub-pipelined design of the AES in CMOS 180 nm. In *5th international conference on design and test workshop* (*IDT*), (pp 31–36).
6. Chang, C. J., Hu, C. W., Chang, K. H., Cheng Y. C., Hsieh, C. C. (2008). High throughput 32-bit AES implementation in FPGA. *IEEE Asia Pacific conference on circuits and systems* (*APCCAS*), (pp. 1806–1809).
7. Li, H., & Friggstad, Z. (2005). An efficient architecture for the AES mix columns operation. *IEEE International Symposium on Circuits and Systems, 5*, 4637–4640.
8. Samiee, H., Atani, R. E., & Amindavar, H. (2011). A novel area-throughput optimized architecture for the AES algorithm. In *International conference on electronic devices, systems and applications* (*ICEDSA*), (pp. 29–32).
9. Reddy, S. K., Saktivel, R., & Paneeth, P. (2011). VLSI implementation of AES crypto processor for high throughput. *International Journal of Advanced Engineering Sciences and Technologies, 6*, 22–26.
10. Zhang, X., & Parhi, K. K. (2004). High speed VLSI architectures for the AES algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) systems, 12*(9), 957–967.
11. Kamal, A. A., & Youssef, A. M. (2008). An area optimized implementation of the advanced encryption standard. In *International conference on microelectronics* (pp 159–162).
12. Hammad, I., El-Sankary, K., & El-Masry, E. (2010). High speed AES encryptor with efficient merging techniques. *IEEE Embedded Systems Letters, 2*(3), 67–71.
13. Fan, C. P., & Hwang, J. K. (2008). FPGA implementations of high throughput sequential and fully pipelined AES algorithm. *International Journal of Electrical Engineering, 15*(6), 447–455.

14. Swankoski, E. J., Brooks, R. R., Narayanan, V., Kandemir, M., & Irwin, M. J. (2004). A parallel architecture for secure FPGA symmetric encryption. In *Proceedings in 18th international symposium on parallel and distributed processing symposium* (p. 132).
15. Yoo, S. M., Kotturi, D., Pan, D. W., & Blizzard, J. (2005). An AES crypto chip using a high speed parallel pipelined architecture. *Microprocessors and Microsystems, 29*, 317–326.
16. Granado Criado, J. M., Vega Rodriguez, M. A., Sanchez Perez, J. M., & Gomez Pulido, J. A. (2010). A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. *Integration the VLSI Journal, 43*, 72–80.
17. Jyrwa, B., & Paily, R. (2009). An area-throughput efficient FPGA implementation of block cipher aes algorithm. In *International conference on advances in computing*, *control and telecommunication technologies* (pp. 328–332).
18. Choi, H. S., Choi, J. H., & Kim, J. T. (2008). Low power AES design using parallel architecture. In *International conference on convergence and hybrid information technology*, (pp. 413–416).
19. Li, H., & Li, J. (2005). A high performance sub-pipelined architecture for AES. In *IEEE International conference on computer design: VLSI in computers and processors* (*ICCD*), (pp. 491–496).
20. Luo, A. W., Yi, Q. M., & Shi, M. (2011). Design and implementation of area-optimized AES based on FPGA. In *International conference on business management and electronic information* (*BMEI*), (pp. 743–746).
21. Satoh, A., Morioka, S., Takano, K., & Munetoh, S. (2001). A compact Rijndael hardware architecture with S-box optimization. *ASIACRYPT, 2001*, 239–254.
22. Anitha Christy, N., & Karthigaikumar, P. (2012). FPGA implementation of AES algorithm using Composite Field Arithmetic. In *International conference on devices*, *circuits and systems* (*ICDCS'12*), (pp. 713–717).
23. Paar, C. (1996). A new architecture for a finite field multiplier with low complexity based on composite fields. *IEEE Transactions on Computers, 45*(7), 856–861.
24. Zhang, X., & Parhi, K. K. (2006). On the optimum constructions of composite field for the AES algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs, 53*(10), 1153–1157.
25. Savas, E., & Koc, C. K. (1999). Efficient methods for composite field arithmetic. Technical Report, Oregon State University (pp. 1–18).
26. Mathew, S. K., Sheikh, F., Kounavis, M., Gueron, S., Agarwal, A., Hsu, S. K., et al. (2011). 53 Gbps $GF(2^4)^2$ native composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors. *IEEE Journal of Solid-State Circuits, 46*(4), 767–776.
27. Qu, S., Shou, G., Hu, Y., Guo, Z., & Qian, Z. (2009). High throughput, pipelined implementation of AES on FPGA. In *International symposium on information engineering and electronic commerce*, (pp. 542–550) Ternopil, Ukraine.
28. Huang, J., Susilo, W., & Seberry, J. (2011). Repeated differential properties of the AES-128 and AES-256 key schedules. In *10th international conference on trust*, *security and privacy in computing and communications* (*TrustCom*), (pp. 525–532).
29. Kumari, M. S., Mahesh Kumar. D., & Rama Devi, Y. (2011). High throughput-less area efficient FPGA implementation of block cipher AES algorithm. In *International conference on advanced computing*, *communication and networks*, (pp. 484–489).
30. Qin, H., Sasao, T., & Iguchi, Y. (2005). An FPGA design of AES encryption circuit with 128-bit Keys. In *Proceedings of the 15th ACM Great Lakes symposium on VLSI* (*GLSVLSI*), (pp. 1–5).
31. Thongkhome, K., Thanavijitpun, C., & Choomchuay, S. (2011). A FPGA design of AES core architecture for portable hard disk. In *8th International conference on computer science and software engineering* (*ICSSE*), (pp. 223–228).
32. Zhang, Y., & Wang, X. (2010). Pipelined implementation of AES encryption based on FPGA. In *IEEE conference on information theory and information security* (*ICITIS*), (pp. 170–173).
33. Iyer, N. C., Anandmohan, P. V., Poornaiah, D. V., & Kulkarni, V. D. (2006). High throughput, low cost, fully pipelined architecture for AES crypto chip. In *India conference*, *annual IEEE*, (pp. 1–6).

Dr. P. Karthigaikumar received his Bachelor of Engineering degree in Electrical and Electronics Engineering from the Bharathiar University, India in 1999 and his Master of Engineering degree with Distinction in Applied Electronics from Bharathiar University, India in 2002. He completed Ph.D degree in Information and Communication Engineering under Anna University, India in 2011, focusing on FPGA and ASIC implementation of Media Security processor. He is a member of IEEE (MIEEE), senior member of Association of Computer Electronics and Electrical Engineers (ACEEE), member of International Association of Engineers (MIAENG) and member of International Association of Computer sciences and Information Technology (MIACSIT). He joined Karunya University; Coimbatore, India in 2000 and worked for 13 years. He is now Professor and Head in Electronics and Telecommunication Engineering in Karpagam College of Engineering, Coimbatore, India. He has published more than 60 papers in in journals and conferences. He received IETE K S Krishnan award for the best system oriented research paper in the year 2010. He applied for 2 Indian patent and is published in Indian Patent Journal. He is a reviewer for different reputed journals like Elsevier, Wiley, Inderscience etc., and he has been the Guest editor for few special issues in Hindawi, Elsevier, Inderscience, Springer. His research interest includes FPGA implementation of Media security algorithm and Signal Processing algorithm.

N. Anitha Christy received the B.E degree in Electronics and Instrumentation Engineering from Sri Ramakrishna Engineering College, Anna University, Coimbatore, Tamilnadu, India, in 2010, M.Tech in Applied Electronics, Karunya University, India in 2012. She has 5 Publications to her credit. Her research interests includes Cryptography and Digital image Processing.

Dr. N. M. Siva Mangai received her Bachelor of Engineering degree in Electronics and Communication Engineering with distinction from the Madurai Kamaraj University, India in 2000 and her Master of Engineering degree in VLSI Design from PSG College of Technology, Bharathiar University, India in 2002. She completed her Ph.D degree in Information and Communication Engineering under Anna University, Chennai, India in 2011, focusing on Power optimization and failure detection techniques for memory. She is a member of VLSI Society of India (VSI), International Association of Engineers (IAENG), International Association of Computer Science and Information Technology (IACSIT), ICGST Academic Community and The Society of Digital Information and Wireless Communications (SDIWC), Institute of Doctors, Engineers and Scientists (IDES). She is currently working as Associate Professor in Electronics and Communication Engineering, Karunya University, Coimbatore, India.