Int J Parallel Prog (2018) 46:904–922
https://doi.org/10.1007/s10766-017-0545-7

High Performance GCM Architecture for the Security of High Speed Network

Vanitha Mohanraj[1] · R. Sakthivel[1] ·
Anand Paul[2] · Seungmin Rho[3]

Received: 4 July 2017 / Accepted: 3 December 2017 / Published online: 23 December 2017

Abstract Advanced Encryption Standard (AES) is an effective cryptography algorithm for providing the better data communication since it guaranties high security. The Galois/Counter Mode (AES-GCM) has been integrated in various security constrained applications because it provides both authentication and confidentiality. AES algorithm helps to provide data confidentiality while authentication is provided by a universal GHASH function. Since most of existing GCM architectures concentrated on power and area reduction but an compact and efficient hardware architecture should also be considered. In this paper, high-performance architecture for GCM is proposed and its implementation is described. In order to achieve higher operating frequency and throughput, pipelined S-boxes are used in AES algorithm. For a GCM realization of AES, a high-speed, high-throughput, parallel architecture is proposed. Experimental results proves that the performance of the proposed work is around 17% higher than the existing architecture with 3 Gb/s throughput using TSMC 45-nm CMOS technology.

Keywords Advanced Encryption Standard · Galois/Counter Mode · GHASH function · Parallel architecture · High performance

✉ Anand Paul
paul.editor@gmail.com

[1] Vellore Institute of Technology, Vellore, India

[2] The School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea

[3] Department of Media Software, Sungkyul University, Anyang, South Korea

1 Introduction

The modern digital world demands the need for secured data transmission in every walks of the wired and wireless communications and transactions. The RFID based secured data authentication has a challenge of using low power consuming cryptographic integrated circuits. The implementation of wireless WiFi, WiMax, WLAN protocol and IoT applications involves a challenge of implementing the protocol with less area, low power consumption with high throughput [1]. It is difficult to obtain both data confidentiality and authenticity simultaneously for the applications like high speed network [2]. Moreover the computation cost of the high speed networks is very high. Many modes of process are available to upturn the speed. Common modes of block cipher algorithms are Cipher Feed Back (CFB), Electronic Code Book (ECB), Cipher Block chaining (CBC) and Output Feedback (OFB). The architecture of all the above mentioned has data dependency problem. We can propose a parallel architecture only if the data dependency is not available in the architecture else we could only go with serial mode of data processing [3]. So it is difficult to parallelize the code. Counter (CTR) mode which helps to increase the security has no data dependency, so pipelining and parallel processing can be attained efficiently, which increases the algorithm performance. Similar to other modes, Galois Counter Mode of Operation (GCM) is the recently proposed mode which is capable of processing data at high speed by implementing parallelism. Hardware implementation is very hard for the attacker to hack the code. So, it is better to implement it on the hardware to achieve the high performance with reduced latency and less cost. GCM has individuality in which processing the data can be sequential or in parallel. The useful property of GCM is that it accepts arbitrary length of initialization vector which helps to improve the security.

Experimental results of AES-GCM are used in various applications such as Ethernet, Wireless Local Area Network (WLAN), remote database, file system and network storage etc. In GCM the confidentiality of data is achieved by AES and authentication is achieved by the GHASH function. AES algorithm can be implemented depending on the security level and speed required for the application. GHASH function can also be implemented in different ways to obtain area efficient or high performance hardware.

Vliegen et al. [4] works on maximizing the throughput of side-channel-protected AES-GCM implementations on an FPGA and they obtained the throughput of 15.24 Gbit/s.

Paul et al. [5] works on the development of efficient, particle swarm based retiming and hardware architecture development for AES-GCM. Koteshwara [6] discuss about the development of new security schemes for light weight cryptography algorithms.

The remaining portion of this paper is given as follows. Section 2 presents the related work carried out in this area. Next, Sect. 3 gives different modes of operations for implementing the block cipher algorithm. Section 4 presents the AES-GCM architecture. Section 5 describes about the proposed architecture. Section 6 shows the experimental results. Section 7 gives the analysis of simulated results. Section 8 presents mathematical analysis. Section 9 presents the cryptanalysis of the AES architecture and Sect. 10 concludes this work.

2 Related Work

Different types of GCM architectures are explained by various authors. Hori et al. proposed the dynamic partial reconfiguration (DPR) system for AES. In this architecture, only specific portion of the circuit has been updated with another circuit and the remaining portion of the circuit remains functioning. Zhenhen implemented the AES algorithm using ECB mode. Satoh et al. [7] designed the high throughput architecture using CTR mode which is been used for encrypting the plain text and hash function used to perform multiply and add operation over the GF (2^{128}) to generate the tag value. Four different types of multipliers are used parallel multiplier, sequential multiplier and Multi block multiplier and pipelined multiplier [8].

Mehran et al. evaluated the performance of more than 40 Sboxes utilizing the 65-nm CMOS technology. To design the least complexity Sbox the formulation for the Galois Field inversion in GF (2^4) has been applied and analysed the peak power consumptions of the Sboxes by allowing the switching operations. For attaining low latency of AES-GCM mode high speed architectures has been proposed and q-adder and parallel adder multiplier has also been used. Some of the existing multipliers are bit parallel multiplier, digit serial, Karatsuba multiplier, bit parallel multiplier and GF (2^{128}) multiplier. Each of the multiplier varies with different critical path delay.

Fault tolerant architecture for AES processor has been designed by An et al. [9]. They have used two different approaches like Triple Modular Redundancy approach and Triple Temporal Redundancy approaches and they modelled the Sbox using Configurable Sbox array (CSBA). From the implementation they identified that the multiplicative inverse in GF(2^4) requires more XOR gates. So, Kumar et al. [10] used Multiplexers (MUX) to realize the XOR gate to decrease the delay and area of the architecture.

Abhiram et al. [11] implemented the Sbox using look up table. They dynamically changed the Sbox value at run time. Dual key AES has been introduced for increasing the security instead of using the same key. The clock frequency was set to one GHz clock frequency and output was obtained in 40 clock cycles. Kasper and Schwabe [12] proposed bit sliced implementation on the GCM architecture in order to protect against the timing attack. Usually, the authentication tag is calculated using a sequential hardware. It achieves considerable area efficiency but it will take so many number of clock cycles that is equal to 128 bit text blocks to find out the tag. Hence it is not suitable for high throughput data communication. Therefore to achieve high performance, a parallel implementation has proposed by McGrew and Viega [13] which performs the GF (2^{128}) multiplications in parallel so that parallel GF (2^{128}) multiplications were performed concurrently. A high performance implementation of GHASH function was proposed by Meloni et al. [14] which can provide authenticity for bigger data blocks in high speed network communication applications. But this type of architecture faces area overhead. There are few works carried out by Kumar et al. [10,15] concentrating only on the SBox and mixed column design because it decides the speed and throughput of the system. Taha et al. discussed that the real threat of Side Channel Attack lies in the capability to mounting attacks over small parts of the key. References [16–23] include various modern forms of GPU implementation and other intelligent methods used in the codomain of toward to scope of this article.

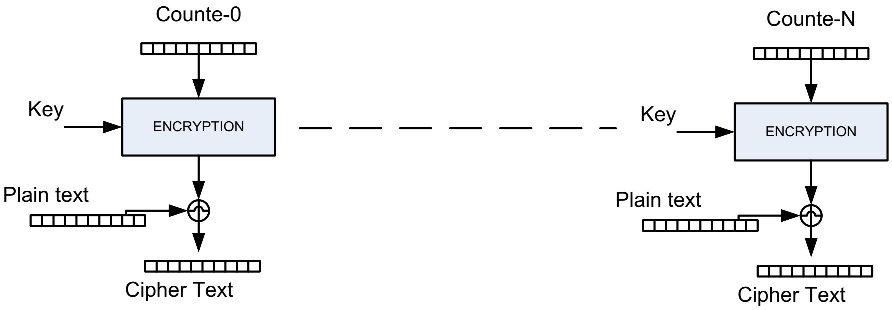


Fig. 1 CTR mode

3 Modes of Operation

Many modes are available but most of the works on AES considers ECB mode of design. But ECB mode is not secure in some cases. The CBC, CFB and OFB offers better security than ECB mode. The encryption speeds in CBC, CFB and OFB modes is much slower than ECB mode. All the modes discussed above have data dependency. So parallel processing cannot be performed. CTR mode which helps to increase the security and has no data dependency. In order to increase the performance of the architecture, pipeling method is used.

3.1 CTR Mode

In counter mode, counter value is encrypted and then XORed the output with the plain text to acquire the cipher text. Decryption applies same procedure to retrieve the plain text. Since the counter value has no data dependency, pipelining can be done efficiently. Figure 1 clearly shows that the CTR mode of operation can be executed in parallel.

3.2 Galois Counter Mode (GCM)

Galois/Counter Mode (GCM) mode was developed by John Viega and David A. McGrew as an enhancement to Carter–Wegman Counter (CWC). NIST in 2007 announced the release of special publication 800-38D and recommended for block cipher modes of Operation such as GCM and Galois Message Authentication Code (GMAC) and made these as official standards. GCM is based on the universal hashing in the finite field GF (2^{10}). A 128-bit is decrypted in twelve clock cycles and additionally it takes ten clock cycles to calculate the authentication tag. Among different modes of operation, GCM mode gives the best result. GCM mode helps to improve the efficiency and performance of the algorithm. Throughput for the communication channel can be achieved with reasonable hardware resources. GCM is defined for 128 block cipher. The GMAC and GCM operation accepts the Initialization Vector (IV) of arbitrary length. Using GCM mode, full advantage of parallel processing can be achieved whereas the other modes like CBC gain significant pipeline stalls that hamper its performance.

GCM combines counter mode with the Galois mode. The significant feature of GCM is that the Galois field multiplication which helps to easily compute the value in parallel. In polynomial, the GF (2^{128}) field used is defined as mentioned in Eq. 1

$$x^{128} + x^7 + x^2 + x + 1 \tag{1}$$

The authentication tag is derived by applying the GHASH function to the input data and it is shown in Eq. 2

$$\mathrm{GHASH}(H, A, C) = X_{m+n+1} \tag{2}$$

where 'H' is the Hash Key, 128 zero bits are encrypted using the cipher algorithm, 'A' is a data which is only authenticated but not encrypted, 'C' is the cipher text data, 'm' is the number of 128 bit blocks in plain text, 'n' is the number of 128 bit blocks in cipher text. The variable X_i for $i = 0, \ldots, m + n + 1$ is defined as mentioned in Eq. 3 which is given by McGrew et al. (2005).

$$X_i = \begin{cases} 0 & \text{for } i = 0 \\ (X_{i-1} \oplus A_i) \cdot H & \text{for } i = 1, \ldots, m-1 \\ (X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H & \text{for } i = m \\ (X_{i-1} \oplus C_{i-m}) \cdot H & \text{for } i = m+1, \ldots, m+n-1 \\ (X_{m+n-1} \oplus (C_n^* \| 0^{128-u})) \cdot H & \text{for } i = m+n \\ (X_{m+n} \oplus (\mathrm{len}(A) \| \mathrm{len}(C))) \cdot H & \text{for } i = m+n+1 \end{cases} \tag{3}$$

Figure 2 explain the general operation of GCM mode. Initialization Vector (IV) is first encrypted, the output value and the plain text is XORed to form the cipher text value. The cipher text value is again XORed with $\mathrm{mult}_{\mathrm{H}}$ and Auth Data_1 to form the another $\mathrm{mult}_{\mathrm{H}}$, which is forwarded for the next parallel data. The authenticated tag value will be generated at the final step. Both the sender and the receiver tag value will be compared; if both are same it proves the authentication and confidentiality.

4 AES-GCM Architecture

The AES is a symmetric block cipher that has been approved by National Institute of Standards and Technologies (NIST). AES is capable of providing more security than Data Encryption Standard (DES). Based on the security level needed by the designer, the key size can be chosen as 128, 192 or 256 bit. But the data size of 128 bits remains constant. The size of the key determines the amount of rounds performed in the algorithm which varies like 10, 12 or 14 rounds. Each round of AES algorithm comprises of four essential transformations. They are Subbytes transformation, Shiftrow, Mixcolumns and finally AddRoundKey. In the final step MixColumn operation won't be performed.

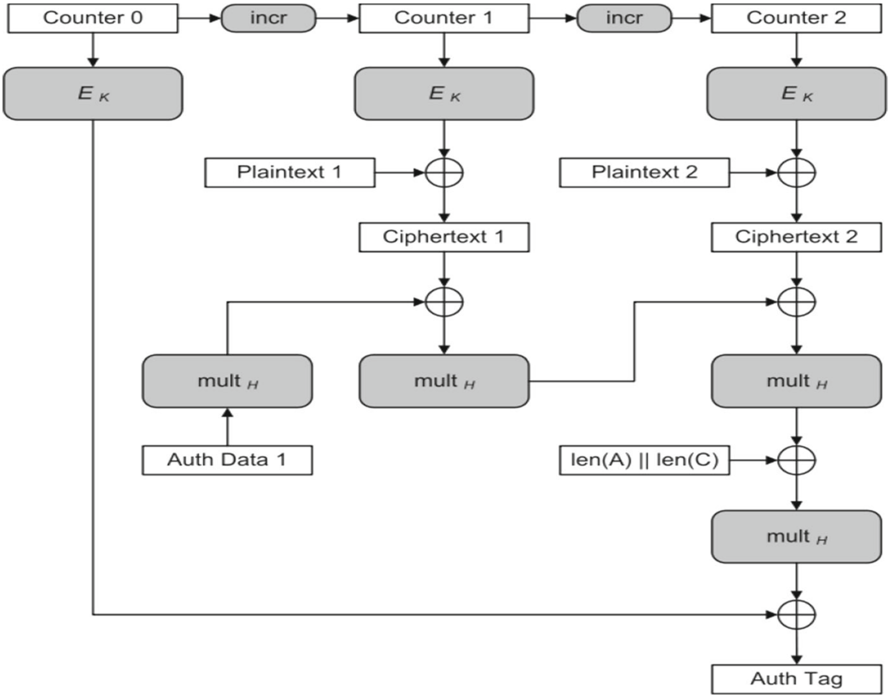


Fig. 2 General GCM mode

The 128 bit input data block will be distributed into sixteen eight bit arrays called state array. During SubBytes transformation all these sixteen blocks will be substituted using Sbox. This operation consists of calculating the multiplicative inverse followed by an affine transformation which converts one vector space into another vector space, which is discussed by the author Satoh et al. [24]. SubBytes transformation is the most area consuming portion of AES algorithm. Sbox can be implemented using look up table by storing the substitution values for all the possible combinations, which can be suitable for FPGA applications deliberated by Zhou et al. [25] and Yang et al. [26]. For Sbox construction it requires 256 × 8 bit ROMs. Another method of implementing Sbox called look up table method, which is purely concentrated on high speed but not on the area, which is not suitable for ASIC implementations as proved by Hodjat and Verbauwhede [27]. Sbox which has been realized using logical gates is called composite field Sbox. So, this method of implementation occupies less area at the same time it gives compromising speed. It is more suitable to implement pipelining technique to increase the operating frequency.

Shift row transformation is one of the operations in AES algorithm discussed by NIST as shown in Fig. 3 and cyclic left shift operation is taking place in this transformation. For the first row array, there is no shifting. Remaining rows except first row are left shifted by one, two, and three times at regular intervals. This transformation does not require any hardware logic and it requires simple wires to route the input to output.

$$\begin{bmatrix} \mathbf{a}_{0,0} & \mathbf{a}_{0,1} & \mathbf{a}_{0,2} & \mathbf{a}_{0,3} \\ \mathbf{a}_{1,0} & \mathbf{a}_{1,1} & \mathbf{a}_{1,2} & \mathbf{a}_{1,3} \\ \mathbf{a}_{2,0} & \mathbf{a}_{2,1} & \mathbf{a}_{2,2} & \mathbf{a}_{2,3} \\ \mathbf{a}_{3,0} & \mathbf{a}_{3,1} & \mathbf{a}_{3,2} & \mathbf{a}_{3,3} \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{a}_{0,0} & \mathbf{a}_{0,1} & \mathbf{a}_{0,2} & \mathbf{a}_{0,3} \\ \mathbf{a}_{1,1} & \mathbf{a}_{1,2} & \mathbf{a}_{1,3} & \mathbf{a}_{1,0} \\ \mathbf{a}_{2,2} & \mathbf{a}_{2,3} & \mathbf{a}_{2,0} & \mathbf{a}_{2,1} \\ \mathbf{a}_{3,3} & \mathbf{a}_{3,0} & \mathbf{a}_{3,1} & \mathbf{a}_{3,2} \end{bmatrix}$$

Fig. 3 Shift row transformation

Fig. 4 Mix column

$$\begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{2} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$\begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{2,0} \\ a_{3,0} \end{pmatrix} \oplus \; Sbox \begin{pmatrix} a_{3,3} \\ a_{1,3} \\ a_{2,3} \\ a_{0,3} \end{pmatrix} \oplus rcon = \begin{pmatrix} b_{0,0} \\ b_{1,0} \\ b_{2,0} \\ b_{3,0} \end{pmatrix}$$

Fig. 5 Round key generation

Followed by a shift row mixcolumn transformation is performed. This operation is performed over Galois Field $3x^3 + x^2 + x + 2$ every column of the array will undergo multiplication and then reduction operation on the polynomial $x^4 + 1$. The multiplication and reduction operation is reduced to a single matrix. Each 32 bit column shown in Fig. 4 shows the multiplication matrix used for this transformation. Using the polynomial it is split into four column blocks and multiplied with 4x4 matrix which combines the operation. Multiplication and addition are performed over GF (2^8) since all the elements are 8 bits. In round key operation, the outputs from the MixColumn transformation is XORed with the key which has been generated in the key expansion module. These keys are known as round keys and each round key will be XORed at the end of each round.

The round keys can be computed at each round. Depending on the key length, the task of finding the round key is slightly different but the logic remains same. For 128 bit key, the last column in the state array is rotated and substitution operation is done by Sbox. After that, 32 bit round constant (rcon) value is XORed with the Sbox output array. The rcon value is calculated based on the Eq. 4.

$$rcon\,(i) = x^{254} + i \; mod \; x^8 + x^4 + x^3 + x + 1 \tag{4}$$

which is performed over GF(2^8). First column of the next round key is calculated using rcon, once it is calculated the remaining three columns of the round key will be calculated by simply XORing of the state array as shown in Fig. 5. Considering the previous round key as the cipher input for the succeeding round key, remaining round keys will be created. Server et al. (2004) proposed a module for key generation which will be generated on the fly.

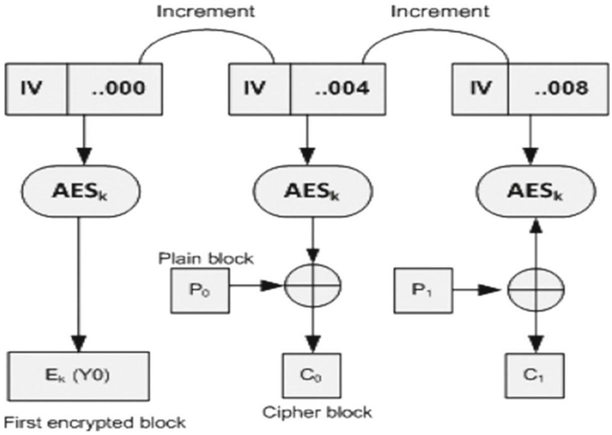


Fig. 6 GCM architecture for AES

Authentication and confidentiality is the main goal of cryptography and there are several techniques to achieve the security. Data confidentiality is the process of providing assurance that the data can be readable only by intended persons. Authentication helps to prevent the data from unauthorized or accidental modification.

In the existing ways like EAX mode by Bellare et al. [28] and Counter with CBC-MAC (CCM) mode by Yang et al. [26] it is evident that the confidentiality and authenticity may be achieved by two separate steps which may neither be pipelined nor enforced in parallel. These methods are therefore cannot be integrated with high speed networks. In AES-GCM, combined authentication and encryption scheme is fused, and this has been used in number of recent standards by NIST and the IEEE. The AES-GCM mode allows fulfilling the industry needs as it is capable of working at very high speeds. Inputs to GCM are Plaintext (P), AES encryption key (K), Additional Authenticated Data (Auth Data), Initialization vector (IV) and the cipher text (C). The Auth Data is used for providing additional data security. The plain text P is split into 128 blocks and if the last blocks are less than 128-bit, zeros are expanded to make it 128 bit length. The Eq. 5 which is mentioned below explains the AES-GCM operation. For simplicity it is assumed that the length of Auth Data and P is exactly divisible by 128. $\mathrm{Y_i}$ gives the counter values for each AES block. Architecture of AES-GCM is provided in Fig. 6. This block speaks about the initialization Vector (IV) being the input for AES block and the whole bit stream is divided into 8 bit blocks and all the blocks works in parallel. For every block, 2^{n} initialization Vector bits are passed to AES block. The encrypted data is being given after the data is being processed through this architecture.

$$\left.\begin{array}{l} Y_0 = IV \\ Y_i = Y_i + 1 \\ H = AES_k\left(0^{128}\right) \\ E_k\left(Y_0\right) = AES_k(Y_0) \\ C_i = P_i\, AES_k\left(Y_i + 1\right) \\ TAG = GHASH\ \left(H, A, C\right) \oplus E_k\left(Y_0\right) \end{array}\right\} \tag{5}$$

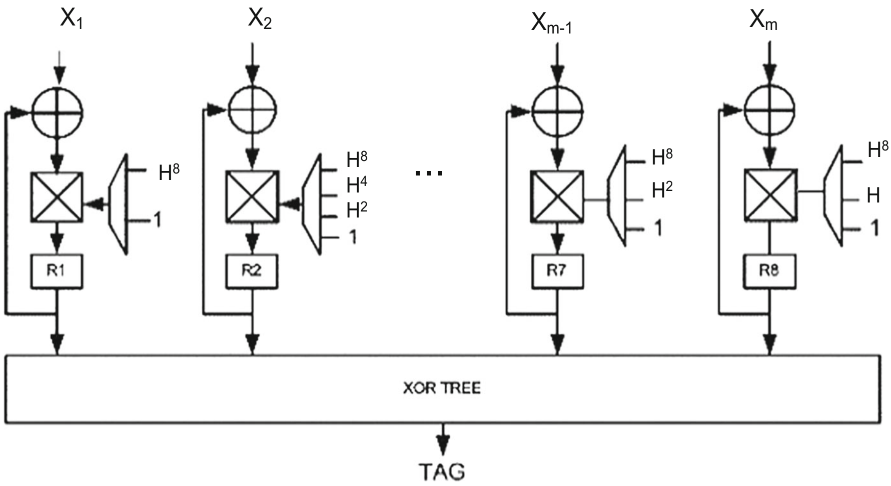


Fig. 7 Parallel architecture for GHASH

Equation 5 shows the sequence of computation that need to be performed to get the authentication tag. Initialization vector *IV* has 96-bit data and it is padded with thirty one zeros followed by a one to make a 128 bit sequence. This will be given to different counter blocks to produce unique sequences. The 128 bit from each block will be XORed with 128 bit plain text to produce cipher text *C*.

The output of 128 bit encrypted blocks is given to GHASH function and finally TAG will be produced as shown in Fig. 7. The E_K value which is obtained by giving Y_0 as the input to the AES block is used to find out the TAG value. The GHASH function module mainly consists of a 128 bit polynomial multiplication value. The polynomial can be written as $x^{128} + x^7 + x^2 + x + 1$. The input to the multiplier is H, which is obtained by performing AES algorithm to 128 bits of initial zero padded value. The other multiplicand can be 128 bit authenticated data, cipher text or length block. Length block value was generated using the len() function. Depending on the length of authenticated data and plain text, the length block produces a 64 bit value. The two 64- bit results are concatenated to form length block i.e. len(A)|| len(P).

The computed value of $E_K(Y_0)$ is passed to the parallel architecture for GHASH function as shown in Fig. 7. The 128 bit Authenticated data blocks followed by 128 bit data blocks are given to GF (2^{128}) multiplier. Another input to the multiplier would be H. After processing the Authenticated data with data blocks, the length of the block is fed into the GF multiplier. The 128 bit $E_K(Y_0)$ value which is obtained by encrypting the first counter block value is XORed at the end to get the final value of TAG. The GHASH function calculates the value by using the Eq. 6.

$$X_1 \bullet H^m \oplus X_2 \bullet H^{(m-1)} \oplus \cdots \oplus X_{(m-1)} \bullet H^2 \oplus X_m \bullet H \tag{6}$$

where *X* is the 128 bit block, *m* is the total amount of blocks.

5 Proposed Architecture

A composite field Sbox is preferred over look up table Sbox as it consumes less area. Depending on the coefficients like v $\in$ GF(2^4), u $\in$ GF(2^2), complexity will change, the irreducible polynomials are (u^2+u+v) and ($\mu^2+\mu+v$) respectively. The constants are v $\in$ GF(2^4) then u $\in$ GF(2^2) over GF $((2^2)^2)/v^2 + v + \phi$ and GF $(((2^2)^2)/(x^2 + x + 1))$ for the composite fields. To reach the optimum design of Sbox $\phi = \{11\}$ and $v = \{1010\}$ are selected. Figure 8 shows the overall methodology of the proposed work. The decision of ASIC or FPGA based methodology is decided by the requirement of highly optimized, huge volume manufacturing or prototype of the proposed model with less optimization respectively. This work proposes the parallel architecture to achieve high performance for the AES-GCM mode. This architecture will progress the latency and the throughput of the architecture presented by Mozaffari Kermani and Reyhani-Masoleh [29] for GHASH. To find the powers of H, GF (2^{128}) multipliers are used. Parallel architecture for GHASH function is capable of calculating the authentication tag within four clock cycles for a data length of 2048 bits. Figure 9 shows the proposed high performance parallel GCM architecture with resistance against scan attack. For increasing the frequency and throughput, a ten round sub pipelined AES module is used. Register R1 to R3 are the sub-pipeline register helps to increase the speed [30].

Initialization vector of 96-bit is given as the input and 32 bits 0^{31} are padded with IV. In GCM mode length of the initialization vector can be of any length. The security can be further increased due to this the unique sequence to the input for the AES blocks, counters are used so that the sequence won't be same. The counters will be incremented for every subsequent block. The 128 bit sequence obtained from the AES blocks will be XORed with 128 bit input blocks to obtain the encrypted data. Then the 128 bit encrypted data bits are fed into the GHASH block. Finally authentication tag will be produced and the encrypted data will be sent along with the tag. Similarly in the decryption side, the encrypted data is XORed with the parallel AES output sequences to retrieve the actual data. Encrypted data is given as an input to the GHASH block. Finally the calculated authentication tag will be compared with the received tag. If both are same, the authentication will be successful and the data will be transferred. On the other hand, if there is any miss match with the tag, the authentication fails. The GCM operates on 128 bits in parallel and inside the AES process the Sbox is made up of composite field. The proposed architecture uses counter mode GCM operation.

The initialization vector is XORed with initial key and this data is passed through 10 rounds of AES algorithm. The response of each step is made compact in the response compactor and the final value is passed into the GHASH function block which develops the authentication tag on the sender side. Now the encrypted data is transmitted with an authentication tag to the receiver. These steps are done in the reverse fashion to get the original data at the receiver side. This architecture is resistant to scan attack.

6 Experimental Results

The encryption and decryption block of the proposed architecture was modelled using Verilog HDL. The ASIC implementation of the proposed design is done by synthe-

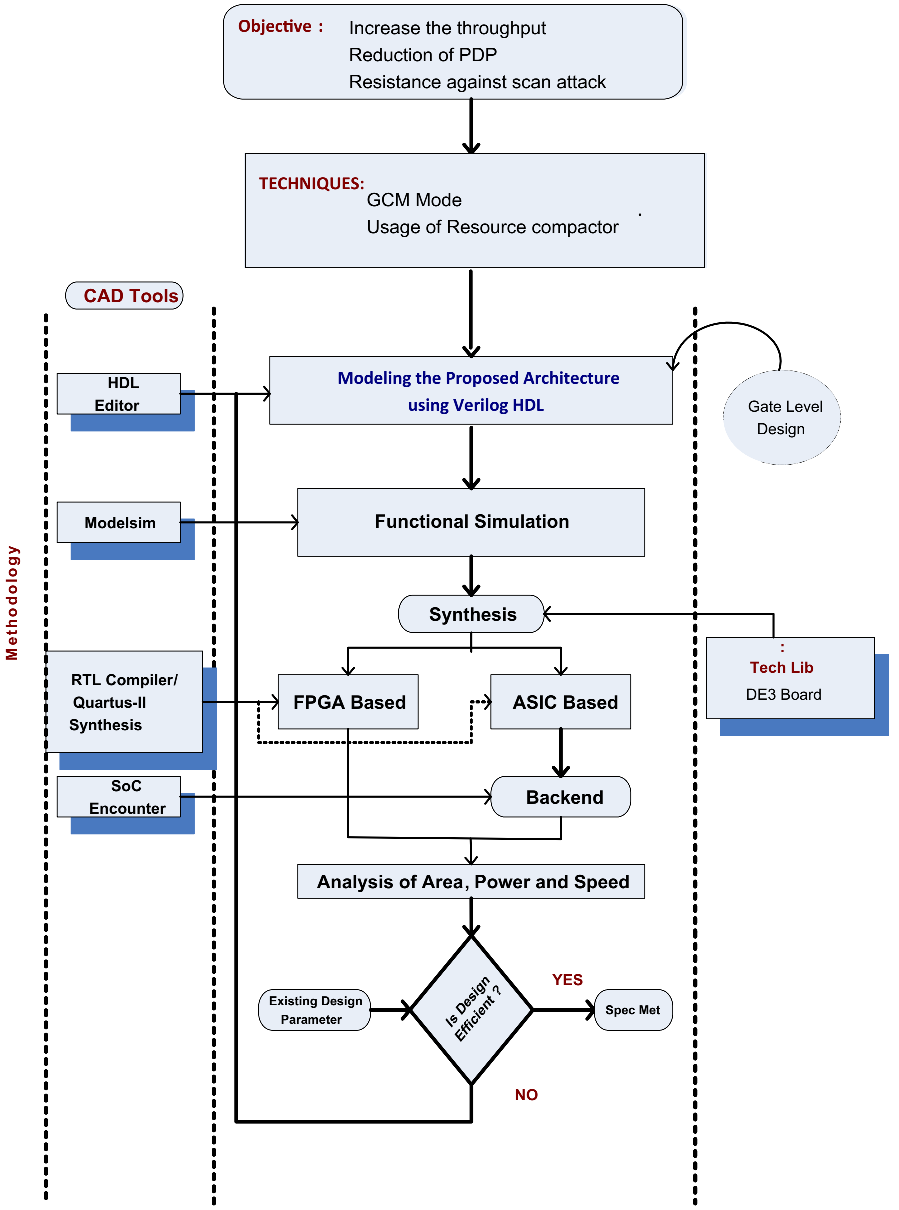

Objective : Increase the throughput
Reduction of PDP
Resistance against scan attack
TECHNIQUES: GCM Mode
Usage of Resource compactor
CAD Tools
Methodology
HDL Editor
Modeling the Proposed Architecture using Verilog HDL
Gate Level Design
Modelsim
Functional Simulation
Synthesis
Tech Lib
DE3 Board
RTL Compiler/ Quartus-II Synthesis
FPGA Based
ASIC Based
SoC Encounter
Backend
Analysis of Area, Power and Speed
Existing Design Parameter
Is Design Efficient ?
YES
Spec Met
NO

Fig. 8 Methodology of implementation

sizing the design with RTL compiler tool of cadence with 90 and 45 nm technology file from TSMC. The synthesized result gives the area, power and timing report of the design and the backend of the design were executed using SOC Encounter of Cadence. The complete layout of the chip of the proposed design after the backend process is shown in Fig. 10 with design specification. Figure 11 shows the comparison

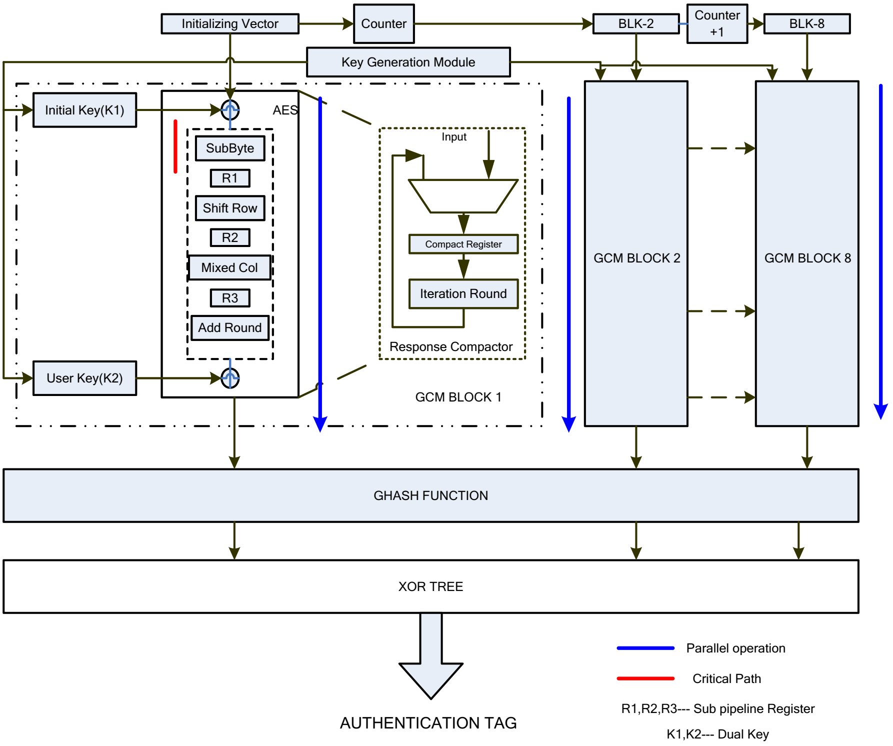


Fig. 9 Proposed high performance parallel GCM architecture

graph of ASIC implementation. The ASIC implementation of the design gives the optimized results. Table 1 compares the design parametric of using normal Sbox with a pipeline Sbox which had been used in our design. The proposed design uses a GF (2^{128}) Multiplier and its area, power and timing are given in the Table 2 for both 90 and 45 nm technology. The cryptanalysis on the proposed architecture was performed; since iterative architecture for compressing the scan chain flip flop have used the in the response compactor.

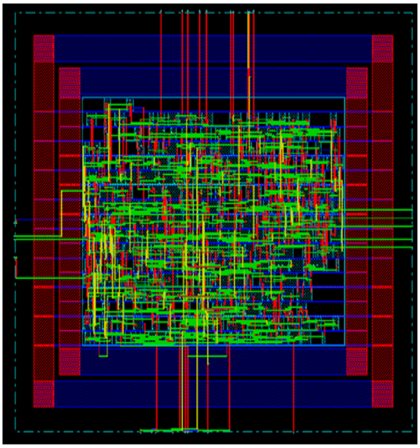
6.1 Power Delay Product (PDP) Calculation

Power delay product is a metric for comparing the design trade-off. The supply voltage $\mathrm{V_{DD}}$ is directly proportional to power dissipation and inversely proportional to timing or delay as shown in Eqs. 7 and 8 [31].

$$P_{SC} = I_{SC} \times V_{DD} \tag{7}$$

$$Critical\ path\ timing = delay = \frac{KC_L}{V_{DD}\left(1 - \frac{V_T}{V_{DD}}\right)^2} \tag{8}$$

Here ‘K’ is the process constant, ‘$\mathrm{V_T}$’ is threshold voltage, ‘$\mathrm{V_{DD}}$’ is supply voltage and ‘$\mathrm{C_L}$’ is the load capacitance faced in the path.



Technology : 90nm (TSMC)

Operating Frequency: 464MHz

Area :5858122μm^2

Total Power : 1.72W

Throughput : 184 Gb/s

Fig. 10 ASIC implementation of proposed architecture

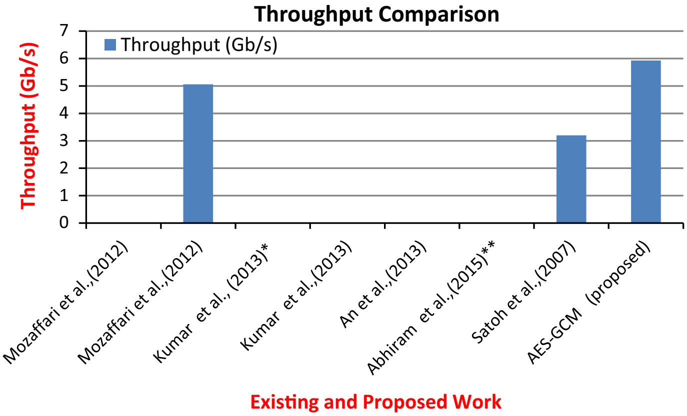


Fig. 11 Comparison graph of ASIC implementation

Table 1 Comparison of normal Sbox and pipelined Sbox using TSMC 45 nm technology

AES (GPDK 45 nm)	Operating frequency (MHz)	Total area (μm^2)	Dynamic power (nW)	Leakage power (nW)	Total power (nW)
Pipelined Sbox	437	534	134,005	18	134,024

From the following formula,

$$\begin{aligned} \textit{Power Delay Product (PDP)} &= \textit{Power} \times \textit{Delay} \\ \text{PDP} &= 1.7\ \text{W} \times (1/464)\ \text{ns} \\ &= 3.66 \times 10^{-3}\ \text{n W s} \end{aligned}$$

The frequency and throughput calculation are done using their equations.

Table 2 Performance of GF (2^{128}) multiplier using TSMC 90 and 45 nm technology

GF multiplier	Operating frequency (MHz)	Total area (μm^2)	Leakage power (mW)	Dynamic power (mW)	Total power (mW)
GF (2^{128}) multiplier GPDK (90 nm)	860	252,133	2.8	49	51.8
GF (2^{128}) multiplier GPDK (45 nm)	322	88,210	0.002	21.38	21.39

7 Simulation and Analysis

High performance AES-GCM architecture has been implemented on 8 blocks and 16 blocks. Every block of data will be executed in parallel AES module and GF (2^{128}) multiplier which can work concurrently. In the existing method, Mozaffari-Kermani and Reyhani Mosoleh et al. [29] used ten round pipelined AES blocks.

Each AES block contains encryption and decryption module of the proposed architecture and a separate key expansion module. In the proposed work a common key expansion module is used for both 8 and 16 blocks to reduce the area complexity. Also sub pipelining is achieved by converting the composite Sbox into pipelined Sbox. The simulation results for a test data input are as follows,

Encryption

Plain text: 2b7e151628aed2a69bf7158809cf4f3c
Initial vector: aaaabbbbccccddddeeeeffff
Key: 2b7e151628aed2a69bf7158809cf4f3c
Encrypted data: 1d7dd722d4e42718ec76bc6bfe30f80aafccfe4d…
Tag: db0657751e6fodb8f837ece939f4075c

Decryption

Cipher data: 1d7dd722d4e42718ec76bc6bfe30f80aafccfe4d…
Initial vector: aaaabbbbccccddddeeeeffff
Decrypted data: 2b7e151628aed2a69bf7158809cf4f3c
Data Authentication: enabled

The design is synthesized using TSMC 90 nm technology. It shows that the existing frequency of 396 MHz which is shown in Table 3. During encryption it consumes an area of 6,333,593 $\mu m^2$and a power consumption of 2.15 W. During decryption it consumes the area of 6,334,569 μm^2 and a power consumption of 2.16 W. The proposed architecture works at a frequency of 464 MHz which is high when compared to the existing work. The encryption block takes an area of 5,856,769 μm^2 and consumes a power of 1.7 W. The decryption process consumes 5,858,122 μm^2 with a power consumption of 1.68 W. The throughput got boosted from 5.06 to 5.96 Gbps and the power delay product has reduced from 5.4 to 3.6. Tables 3 and 4 shows the comparison between two architectures in TSMC 90 nm technology and 45 nm technology respectively. In 90 nm technology fast.lib timing library is used to synthesize and in 45 nm technology slow.lib is used. In 45 nm technology, the proposed architecture achieves

Table 3 AES algorithm comparison using TSMC 90 nm CMOS technology

AES (GDPK 90 nm)	Operating frequency (MHz)	Total area (μm^2)	Total power (W)	Throughput (Gb/s)	Power delay product (W ns)
Mozaffari et al. [29]	396	6,333,593	2.15	5.06	5.4
Mozaffari et al. [29]	396	6,334,569	2.16		
Kumar et al. [10,15][a]	1.92	279.41	62.9	NA	NA
Kumar et al. [10,15]	0.217	NA	2.49	NA	NA
An et al. [9][b]	NA	24241	0.21×10^{-3}	1745.8×10^{-6}	NA
Abhiram et al. [11][c]	0.0422	NA	NA	NA	NA
Satoh [32]	250	38,080	NA	3.20	NA
AES-GCM (proposed)	464	5,856,769	1.7	5.93	3.66

NA Data not available

[a] Results only for multiplicative inverse

[b] Results are only for Sbox

[c] FPGA based design

Table 4 AES algorithm comparison using TSMC 45 nm CMOS technology

AES (GPDK 45 nm)	Operating frequency (MHz)	Total area (μm^2)	Total power (W)	Throughput (Gb/s)	Power delay product (W ns)
Mozaffari et al. [29] encryption	129	2,219,088	0.631	58	97
Mozaffari et al. [29] decryption		2,219,554	0.626		
AES-GCM encryption (proposed)	153	2,049,012	0.616	61	121
AES-GCM decryption (proposed)		2,049,479	0.612		

a frequency of 153 MHz whereas in the previous work it was 129 MHz The area overhead is less in proposed architecture whereas the power consumption is almost same, having an improvement of 3 Gbps in the throughput.

8 Mathematical Analysis

Parallel processing by L stages will increase the throughput and frequency as shown in Equations 9. [31]. If L increases T_{sample} decreases and thereby increasing the frequency of operation and throughput.

Table 5 Mathematical analysis

GPDK 90 nm	Operating frequency(MHz)	Total area (μm^2)	Total power (W)	Throughput (Gb/s)	Power delay product (W ns)
AES-GCM (proposed) (8 blocks—128 bit)	464	5,856,769	1.7	5.93	3.66
AES-GCM (proposed) (16 blocks—256 bit)	464	7,632,867	3.2	10.53	6.89

$$T_{sample} = \frac{T_{clock}}{L} \tag{9}$$

The block delay increases for L stage parallel processing. The parallel processing can also be implemented for the sake of reducing the power consumption by reducing the supply voltage (V_0). The propagation delay (T_{pd}) reduces because the charging capacitance (C_{charge}) in the path reduces as shown in the Eq. 10. V_t is the threshold voltage of the device.

$$T_{pd} = \frac{C_{charge} \times V_0}{K\ (V_0 - V_t)^2} \tag{10}$$

and the power consumption in any CMOS circuit is given by Eq. 11

$$P_{CMOS} = C_{total} \times V_0^2 \times f \tag{11}$$

Proposed architecture has 8 blocks with each processing 16 bits thereby 128-bit input text was encrypted in the same clock cycle. Table 5 demonstrates the frequency, power, area, throughput and PDP. Same architecture can be extended for 256 bits with 16 block sizes.

$$\text{Frequency} = 464\ \text{MHz}$$

From the equation,

$$Throughput = \frac{no.\ of\ bits * frequency}{no.\ of\ rounds}$$
$$\text{Throughput} = (128{*}464)/10 = 5.93\ \text{Gbps}$$

9 Cryptanalysis of Proposed AES Architecture

AES Cryptographic processors are subjected to side channel attack. If the processor uses the intermediate registers, the probability of hacking is easier. The Scan chain based flip-flop used in the design synthesis is exposed to the scan attack. The hackers can easily run the processor for two to three steps in an AES round by operating in

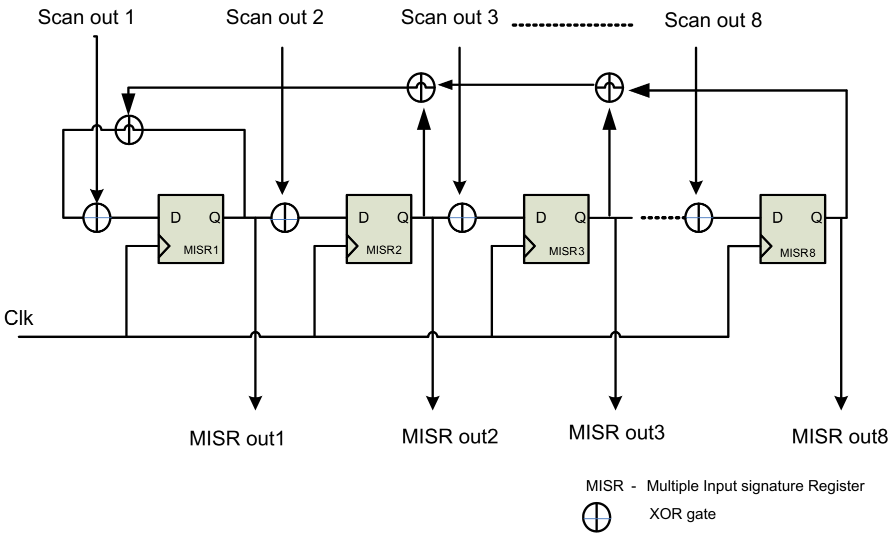


Fig. 12 Proposed response compactor architecture against scan attack

standard mode and then switch over to testing mode which can able to recover the data stored in the intermediate register made of scan based FF. It is possible to hack the data by running the processor for few clock cycles by applying all possible inputs and outputs vectors in a reversal fashion. Based on the hamming distance between the input pairs the plain text can be hacked. Usage of scan based FF cannot be avoided because it facilitates the easy testability of a ICs and this opens an easy way for the hackers to steal the data. Another way to increase the security is to use the concept of dual key which was proposed by Abhiram et al. [11], where the initial key (K1) and used key (K2) has been used to increase the security instead of using one key, as shown in Fig. 9.

A response compactor register was proposed by some of the authors like Ali et al. [33], Ege et al. [34] and DaRolt et al. [35]. They have used response compactor instead of using 10 registers for ten rounds to store the intermediate results. Multiple Input Signature Register (MISR) is a compactor discussed by Mitra et al. [36] which compresses the scan FF intermediate data in a compact register by XOR operation as shown in the Fig. 12. This will reduce the area and simultaneously make the architecture resistant to scan attack. We could see this architecture provide two hamming distance (1 and 8) which are at extreme so that we get $40 \approx 2^{5.31}$ probable values for every key byte. So, the hypotheses for last key is $(25.32)^{16} = 2^{85.15}$ which cannot be brute-forced easily.

10 Conclusion

In this work, a high performance, parallel AES-GCM is implemented. The security of the proposed architecture is increased by using response compactor. The proposed

AES architecture has achieved 17% increase in the throughput by using GCM mode and there is 66% reduction in Power Delay Product (PDP). ASIC implementation in Cadence 90 nm as well as in 45 nm Complementary Metal-Oxide Semiconductor (CMOS) technology shows that the proposed architecture works at higher frequencies, consuming less power and occupies less area compared with the previous work. So the AES-GCM architecture could be incorporated with high performance data transmission to improve the efficiency as well as security of the sensitive data.

Acknowledgements This study was supported by the Next-Generation Information Computing Development Program through National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (2017M3C4A7066010). This study was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (NRF-2017R1C1B5017464).

References

1. Cuomo, S., Michele, P.D., Piccialli, F., Galletti, A., Jung, J.E.: IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario. Expert Syst. Appl. **79**, 101–111 (2017)
2. Chianese, A., Marulli, F., Moscato, V., Piccialli, F.: A smart multimedia guide for indoor contextual navigation in Cultural Heritage applications. In: Proceedings of International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013, (2013)
3. Chianese, A., Piccialli, F.: SmaCH: a framework for smart cultural heritage spaces. In: Proceedings of 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014, pp. 477–484 (2015)
4. Vliegen, J., Reparaz, O., Mentens, N.: Maximizing the throughput of threshold-protected AES-GCM implementations on FPGA. In: 2017 IEEE 2nd International Verification and Security Workshop (IVSW), pp. 140–145. IEEE (2017)
5. Paul, A., Victoire, T.A.A., Jeyakumar, A.E.: Partical swarm approach for retiming in VLSI. In: 2003 46th Midwest Symposium on Circuits and Systems, vol. 3, pp. 1532–1535 (2003)
6. Koteshwara, S., Das, A., Parhi, K.K.: FPGA implementation and comparison of AES-GCM and Deoxys authenticated encryption schemes. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4. IEEE (2017)
7. Satoh, A., Sugawara, T., Aoki, T.: High-performance hardware architectures for galois counter mode. IEEE Trans. Comput. **58**(7), 917–930 (2009)
8. Farina, R., Cuomo, S., De Michele, P., Piccialli, F.: A smart GPU implementation of an elliptic kernel for an ocean global circulation model. Appl. Math. Sci. **7**(61–64), 3007–3021 (2013)
9. An, T., de Barros Naviner, L.A., Matherat, P.: A low cost reliable architecture for S-boxes in AES processors. In: Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), New York, pp. 155–160, USA (2013)
10. Kumar, Saurabh, Sharma, V.K., Mahapatra, K.K.: An improved VLSI architecture of S-box for AES encryption. Proceedings of International Conference on Communication Systems and Network Technologies, Gwalior, pp. 753–756, India (2013)
11. Abhiram, L.S., Sriroop, B.K., Gowrav, L., Punith, K.H., Lakkannavar, M.C.: FPGA implementation of dual key based AES encryption with key based S-box generation. In: Proceedings of International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pp. 577–581, India (2015)
12. Kasper, E., Schwabe, P.: Faster and timing-attack resistant AES-GCM. In: *Proceedings of International Workshop Cryptographic Hardware and Embedded Systems (CHES '09),* Lausanne, pp. 1–17, Switzerland (2009)
13. McGrew, D.A., Viega, J.: The Galois/Counter Mode of Operation (GCM), NIST Modes Operation Symmetric Key Block Ciphers (2005)
14. Meloni, N., Negre, C., Hasan, M.A.: High performance GHASH function for long messages. In: *Proceedings of International Conference on Applied Cryptography and Network Security (ACNS '10),* Beijing, pp. 154–167, China (2010)

15. Kumar, S., Sharma, V.K., Mahapatra, K.K.: Low latency VLSI architecture of S-box for AES encryption. In: Proceedings of International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, pp. 694–698, India (2013)
16. Elliptic Semiconductor Inc.: Ultra-high throughput AESGCMCore-40 Gbps (2008)
17. Wu, H.: On computation of polynomial modular reduction. Technical Report Center for Applied and Cryptographic Research (2000)
18. Helion Technology.: AES-GCM Cores (2007)
19. National Institute of Standards and Technologies: Announcing the Advanced Encryption Standard (AES), Information Processing Standards Publication. No. 197, pp. 1–51 (2001)
20. Vanitha, M., Sakthivel, R., Subha, S.: Highly secured high throughput VLSI architecture for AES algorithm. In:International Conference on Devices, Circuits and Systems(ICDCS), Coimbatore, pp. 403–407, India (2012)
21. Paul, A., Ahmad, A., Rathore, M., Jabbar, S.: Smartbuddy: defining human behaviors using big data analytics in social internet of things. IEEE Wirel. Commun. **23**(5), 68–74 (2016)
22. Paul, A., Daniel, A., Ahmad, A., Rho, S.: Cooperative cognitive intelligence for internet of vehicles. IEEE Syst. J. **11**(3), 1249–1258 (2015)
23. Paul, A.: Real-time power management for embedded M2M using intelligent learning methods. ACM Trans. Embed. Comput. Syst. (TECS) **13**(5s), 148 (2014)
24. Satoh, A., Morioka, S., Takano, K., Munetoh, S.: A compact Rijndael hardware architecture with S-box optimization. In: International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT, Gold Coast, pp. 239–254, Australia (2001)
25. Zhou, G., Michalik, H., Hinsenkamp, L.: Efficient and high-throughput implementations of AES-GCM on FPGAs. In: Proceedings of International Conference on Field-Programmable Technology (ICFPT), Kitakyushu, pp. 185–192, Japan (2007)
26. Yang, B., Mishra, S., Karri, R.: High speed architecture for Galois/counter mode of operation (GCM). In: International Association for Cryptologic Research (IACR), pp. 47–50 (2005)
27. Hodjat, A., Verbauwhede, I.: Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors. IEEE Trans. Comput. **55**(4), 366–372 (2006)
28. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In: Proceedings of Fast Software Encryption(FSE), Delhi, pp. 389–407, India (2004)
29. Mozaffari Kermani, M., Reyhani-Masoleh, A.: Efficient and high-performance parallel hardware architectures for the AES-GCM. IEEE Trans. Comput. **61**(8), 1165–1178 (2012)
30. Piccialli, F., Cuomo, S., De Michele, P.: A regularized MRI image reconstruction based on hessian penalty term on CPU/GPU systems. Procedia Comput. Sci. **18**, 2643–2646 (2013)
31. Parhi, K.K.: VLSI Digital Signal Processing Systems: Design and Implementation, Chap. 3. Wiley, Hoboken (1999)
32. Satoh, A.: High-speed parallel hardware architecture for Galois counter mode. In: International Symposium on Circuits and Systems (ISCAS), pp. 1863–1866, New Orleans, Los Angeles (2007)
33. Ali, S.S., Sinanoglu, O., Karri, R.: AES design space exploration new line for scan attack resiliency. In: Proceedings of IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Playa del Carmen, pp. 1–6, Mexico (2014)
34. Ege, B., Das, A., Gosh, S., Verbauwhede, I.: Differential scan attack on AES with X-tolerant and X-masked test response compactor. In: Proceedings of Euromicro Conference on Digital System Design, Izmir, pp. 545–552, Turkey (2012)
35. DaRolt, J., Natale, G.D., Flottes, M.L., Rouzeyre, B.: Scan attacks and countermeasures in presence of scan response compactors. In: Proceedings of European Test Symposium (ETS), Trondheim, pp. 19–24, Norway (2011)
36. Mitra, S., Mitzenmacher, M., Lumetta, S.S., Patil, N.: X-tolerant test response compaction. Des. Test Comput. **22**(6), 566–574 (2005)