VNU Journal of Science: Comp. Science & Com. Eng., Vol. 37, No. 2 (2021) 32-46

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

Ultra-High-Throughput Multi-Core AES Encryption Hardware Architecture

Pham Khoi Dong[1,2], Hung K Nguyen[2],
Fawnizu Azmadi Hussin[3], Xuan Tu Tran[1,*]

[1]*VNU Information Technology Institute, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[2]*VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[3]*Universiti Teknologi Petronas, Seri Iskandar, Perak, Malaysia,*

Received 17 February 2021
Revised 18 July 2021; Accepted 31 August 2021

Abstract: Security issues are always a big challenge in high-speed data transfer between devices. The encryption in cyber security needs high throughput to meet data transfer rates and low latency to ensure the quality of services. New data transfer standards such as IEEE P802.3bs 2017 stipulate the maximum data rate up to 400 Gbps. However, according to our survey, single-core AES architectures implemented on hardware only reach up to a maximum throughput of 275 Gbps. In this paper, we propose a multi-core AES encryption hardware architecture to achieve ultra-high-throughput encryption. To reduce area cost and power consumption, these AES cores share the same KeyExpansion blocks. Fully parallel, outer round pipeline technique is also applied to the proposed architecture to achieve low latency encryption. The design has been modelled at Register-Transfer-Level in VHDL and then synthesized with a CMOS 45nm technology using Synopsys Design Compiler. With 10-cores fully parallel and outer round pipeline, the implementation results show that our architecture achieves a throughput of 1 Tbps at the maximum operating frequency of 800 MHz. These results meet the speed requirements of future communication standards. In addition, our design also achieves a high power-efficiency of 2377 Gbps/W and area-efficiency of 833 Gbps/mm2, that is 2.6x and 4.5x higher than those of the other highest throughput of single-core AES, respectively.

Keywords: AES, high-throughput, multi-core, cryptography, real-time applications.

1. Introduction

Advanced Encryption Standard (AES) was developed by Belgian cryptographer, Vincent Rijmen and Joan Daemen, was published by the National Institute of Standards and Technology (NIST) in 2001 [1]. AES is a symmetric block

* Corresponding author.
E-mail address: tutx@vnu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.290

cipher that is intended to replace DES as the approved standard for a wide range of applications [2]. In AES, the number of cipher rounds depends on the size of the key. It is equal to 10, 12, or 14 for 128-, 192-, or 256-bit keys, respectively. AES encryption round employsconsecutively four primary operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. The operations of AES encryption algorithm with a 128-bit key size are shown in Figure 1.

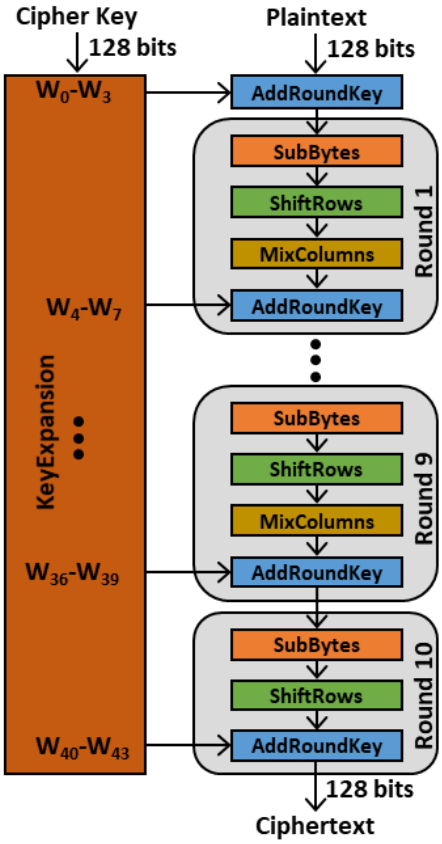

Figure 1. Advanced Encryption Standard Algorithm (AES-128).

Regarding data transfer standards, the IEEE Std 802.3™ was first published in 1985, and then it has been added many functionalities and/or maintenance updates to meet the requirements of applications. Ethernet local area network operation is specified for selected speeds of operations from 1 Mbps to 400 Gbps using a common media access control (MAC) specification and management information base (MIB) [3]. IEEE Std 802.3u added 100 Mbps operation (also called Fast Ethernet), IEEE Std 802.3z added 1000 Mbps operation (also called Gigabit Ethernet), IEEE Std 802.3ae added 10 Gbps operation (also called 10 Gigabit Ethernet), IEEE Std 802.3ah specified access network Ethernet (also called Ethernet in the First Mile), IEEE Std 802.3ba added 40 Gbps operation (also called 40 Gigabit Ethernet) and 100 Gbps operation (also called 100 Gigabit Ethernet). These significant additions are all now included in and are superseded by IEEE Std 802.3-2015 and are not maintained as separated documents [4].

Recently, the IEEE 802.3bs standard, released in 2017, provides a data speed of 200 to 400 Gbps over optical fiber. In addition, other standards that are expected to be published such as IEEE 802.3cm in 2019; IEEE 802.3cn by 2020; IEEE 802.3ck and IEEE 802.3ct in 2021 also have a maximum data rate up to 400 Gbps. Table 1 shows that later standards have higher data transfer rates than previous standards. To meet the requirements of data encryption at data rates up to several hundreds Gbps, it is necessary to use high-speed hardware encryption chips. With a single-core AES-128 architecture using parallel and fully pipeline, the bandwidth is:

$$Throughput = 128 * F_{max} \quad (1)$$

where, F_{max} is the maximum operating frequency of the design. According to Eq. (1), for single-core AES architecture to achieve 400 Gbps speed, the operating frequency of the design must be higher than 3 GHz [5]. In ASIC technology, ICs operating at high frequencies require a complex fabrication process and costly. On the other hand, to increase the operating frequency it is necessary to insert registers into the design at the expense of high area cost and huge power consumption. Therefore, the single-core AES architecture does not meet the speed requirements of new communication standards such as IEEE P802.3bs.

Table 1: Data rates of some IEEE 802.3 standards:

Standard	Date	Description
802.3i	1990	10BASE-T 10 Mbps over twisted pair
803.3j	1993	10BASE-F 10 Mbps over Fiber-Optic
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet at 100 Mbps
802.3z	1998	1000BASE-X Gbps Ethernet over Fiber-Optic at 1 Gbps
802.3ae	2002	10 Gigabit Ethernet over fiber
802.3ba	2010	40 Gbps and 100 Gbps Ethernet
802.3bs	2017	200 Gbps over single-mode fiber and 400 Gbps over optical media

AES implementations can be broadly classified into software and hardware implementation. Compared to software implementation, hardware implementation of AES, by nature, provides more physical security and higher speed [6]. In general, hardware implementation can be performed in either FPGA or ASIC platforms. FPGA provides re-programmable capability and reconfigurability, whereas ASIC suffers from a lack of flexibility. Another difference between ASICs and FPGAs in terms of the performance characteristics is a smaller speed of FPGAs caused by the delays introduced by the circuitry required for reconfiguration. As a result of this speed penalty, any digital circuit implemented in an FPGA is typically slower than the same circuit implemented in an ASIC [7]. Fully parallel, pipelining and sub-pipelining techniques can be employed to increase operational frequency and throughput [8-13]. To achieve high encryption speed, hardware implementation of multi-core AES has attracted more attention [5, 14-17]. On the other hand, AES implementations on the GPU also provide encryption speeds of up to several hundred Gbps [17-20].

Despite high throughput [9-12] due to excessive use of the pipeline, the latency is large and inefficient in terms of hardware resources and power consumption. In real-time applications, latency is an important factor. Delay in encryption, decryption plus other types of delays can affect the quality of the service. Latency in AES encryption is defined by the number of cycles that each data sample takes to go through the encryption data-path before the encrypted output is generated.

$$latency = (number\ of\ pipeline\ stages) \times T_{Clk} = (number\ of\ pipeline\ stages)/f_{max} \quad (2)$$

Although the inner pipeline architecture can increase speed, it also increases the area and latency. When only one outer round pipelining is applied (one stage pipeline per round), the latency is 11 cycles. In the design [8], with two pipeline stages per round, the latency is 21 cycles. For the fully inner and outer round pipelined designs with three or four pipeline stages per round, the latencies are 31 and 41 cycles, respectively. In real-time applications as surveillance cameras, video conferencing, motion and detection, IoT gateways, latency and throughput are critical; therefore, we focus on architecture only outer round pipelining.

In this paper, we propose a hardware AES architecture, which can achieve speed up to hundreds of Gbps. Our design has been modelled at RTL and then synthesized with NanGate 45nm technology using Synopsys Design Compiler. The hardware implementation results demonstrate that our design increases the efficiency of hardware and power consumption. Our two main contributions are as follows:

i) Propose a paralleled multi-core AES architecture that is able to provide ultra-

high-throughput encryption flow. To minimize the design overhead in terms of hardware implementation area and power consumption, KeyExpansion blocks are shared between AES cores.

ii) Encryption latency is the time necessary to encrypt a single block of plaintext. To achieve low latency, outer round pipeline architecture is used in our proposed design. Results of hardware implementation show that our design has lower latency than other related works.

The remaining part of this paper is organized as follows. Section 2 reviews related works to parallel single-core and multi-core AES architectures. Section 3 presents our proposed paralleled multi-core architecture and how it works. In Section 4, we provide details of implementations of the proposed architecture on a 45nm CMOS technology. Finally, Section 5 offers concluding remarks.

2. Related Works

In high-performance hardware designs, pipelining and sub-pipelining techniques can be employed to increase operational frequency and throughput. Hodjat et al. [8] proposed AES-128 core architectures with throughputs of 30 to 70 Gbps corresponding to area cost between 180 and 275 kGates implemented on 180nm CMOS technology. When the design uses the outer round pipelining architecture (one stage pipeline per round), it takes 11 cycles to encrypt a 128-bit block; and therefore, the corresponding latency is 47 ns. To achieve a throughput of 70 Gbps, the authors used a 4-stage pipeline architecture in each round that taking 41 cycles for each 128-bit block, corresponding to a delay of 74.9 ns.

Mathew et al. [9] proposed a reconfigurable AES-128/192/256 encryption engine targeted for the on-die acceleration of real-time encryption/decryption of media content on high-performance microprocessor platforms. The design was fabricated using a CMOS 45nm technology and it achieves a high throughput of 53 Gbps at the maximum operating frequency of 2100 MHz. It spends 55 clock cycles per encryption, so its latency is 26.2 ns.

G. Sayilar and D. Chiou [10] proposed an AES core running at 1000 MHz achieving the highest throughput of 128 Gbps. This architecture has 20 pipeline stages, so needs 20 clock cycles to encrypt a block of data; therefore, latency is 20 ns. Erbagci et al. [12] implemented a fully-unrolled, pipelined AES-128 encryption accelerator using ROM-based S-Boxes in a 65nm bulk CMOS which operates at 2.2 GHz and consumes 523 mW at 1.0 V, 27 ^{o}C. In counter-mode operation (CTR), the throughput is 275.2 Gbps. Although this design achieves high speed, its energy efficiency and area are relatively low (526 Gbps/W and 367 Gbps/mm^2, respectively). The AES architecture in [13] achieves high throughput and low latency using the outer round pipeline technique. This design using single core AES, so throughput is only 111.3 Gbps.

We can observe that the abovementioned proposals do not meet the speed of new communication standards such as IEEE P802.3bs with a throughput of up to 400 Gbps. Moreover, power consumption is an other limitation of those proposed designs. Multi-core AES architectures can address the throughput limitations of single-core AES architectures. On-chip multi-core AES increases the processing throughput and is presented in some works as follows:

The work in [14] using Multicore Processor (AMP-MP) to achieve high throughput and secure Advanced Encryption Standard based on

Counter with Chaining Mode (AES-CCM). The proposed AMP-MP is realized on an 8-bit asynchronous 9-core processor fabricated based on 65nm CMOS process. The experimental results show that the throughput of the authentication is 13.54 Gbps.

Buhrow et al. [5] introduce a new parallelization strategy for Advanced Encryption Standard based on Galois/Counter Mode (AES-GCM). This approach enables the construction of scalable streaming cores that can process multiple separately-keyed packets per clock cycle in wide segmented busses. A demonstration on a single Xilinx Ultrascale FPGA showed that the architecture achieved a throughput of 482 Gbps and the authors also present how the architecture can be used to achieve over 800 Gbps in a system comprising multiple FPGAs. It is possible to implement the design on multi-FPGA systems because the architecture requires no core-to-core communication.

Al-Bahri et al. [21] introduced a parallel implementation of the AES algorithm on a multi-processor platform. This research aims to increase the speed processing of the AES encryption algorithm using the parallel and sequential mechanism. The authors have optimized both the encryption as well as the decryption algorithms. Parallelization in the source code has provided an enormous difference of more than 80% in comparison with the sequential algorithms.

In [15], the authors present an efficient design methodology to implement in reconfigurable hardware devices the GCM combined with the AES for authenticated encryption. They use four AES cores and four binary field multipliers to demonstrate how to break the 100 Gbps speed bound on FPGA. To reduce the critical path of the GHASH operation, they insert four pipeline stages within the GF(2^{128}) multiplication. The final GCM architecture relies on a 4×4 construction and achieves 119 Gbps on Xilinx Virtex-5 devices.

Angelo Barnes et al. [22] implemented the AES encryption algorithm by using multi-core architectures. By using an Intel Core 2-Duo processor (with two single-threaded cores), an Intel Core i3 (with two cores, supporting four threads), and 4x Intel® Xeon® X7560 (with 32 cores, supporting 64 threads), they obtained a highest throughput of 6637 Mbps. This was obtained by running 32 threads on the 32-core machine using the pthread (Posix thread) implementation.

In [23], the authors have designed a configurable AES chip, called AESTHETIC, which enhances security over standard AES designs. The implicit configurability allows the users to switch among AES-extended block ciphers. The chip supports both ECB (Electronic Code Book) and CBC (Cipher Block Chaining) cipher modes with 128-bit, 192-bit, and 256-bit keys. The maximum throughput is around 844.8 Mbps for 128-bit keys, 704 Mbps for 192-bit keys, and 603.4 Mbps for 256-bit keys under an operating frequency of 66 MHz. An on-the-fly key generator is also developed that provides exactly one 128-bit round key per clock cycle. It can be used in not only the extended AES algorithm but also the original AES algorithm.

Despite achieving high throughput, the latency of the designs in [21], [24], and [25] are very high and these designs are inefficient in terms of hardware resources and power consumption due to the excessive use of the pipeline.

For the GPU (Graphics Processing Unit), there have been many works and implementations using the high-throughput parallel computation for the AES algorithm.

Among them, for instance, Ma et al. [26] introduce an implementation based on T-box look-up table, with round keys and T-boxes stored in the shared memory. This work investigates the impact of different input sizes on the overall performance of the GPU. Furthermore, Abdelrahman et al. [17] present an optimization on the AES by accommodating 32 bytes per thread in GTX 1080, achieving a high-throughput speed of 280 Gbps in the ECB encryption. Many works adapted this representation view and proposed the bit-sliced AES implementation [18, 27]. Due to the evolution of GPU hardware, it is now possible to implement the bitsliced AES on different GPUs. Nishikawa et al. [18] proposed a bitsliced ECB-AES implementation on GPU with a different and flexible number of plaintext blocks operated on by a single thread. The work mentioned above roughly achieves 606 Gbps in the configuration where each thread packs four plaintext chunks into eight 64-bit variables referred to as the Bs64. CUDA is a parallel computing platform and programming model developed by Nvidia for general computing on graphical processing units (GPUs). With CUDA, developers can dramatically speed up computing applications by harnessing the power of GPUs [28]. In [19], a high-throughput bitsliced AES implementation is proposed, which builds upon a new data representation scheme that exploits the parallelization capability of modern multi/many-core platforms. The authors use this representation scheme as a building block to redesign all of the AES stages to tailor them for multi-core AES implementation. With this approach, each parallelization unit processes an unprecedented number of thirty-two 128-bit input data. Hence, we can achieve a high order of parallelization by the proposed implementation technique. It leads to a high-throughput CTR and ECB AES encryption/decryption on 6 CUDA-enabled GPUs, which achieve 1.47 and 1.38 Tbps of encryption throughput on Tesla V100 GPU (32-cores), respectively.

In [20], the authors evaluated the throughput and power efficiency of three 128-bit block ciphers on GPUs with recent Nvidia Kepler and AMD GCN architectures. From experiments, the throughput and power efficiency of AES-128 on Radeon DH 7970 (2048 cores) with GNC architecture are 205 Gbps and 1.3 Gbps/W respectively; and those on Geforce GTX 680 (1536 cores) with Kepler architecture are 63.9 Gbps and 0.43 Gbps/W, respectively.

As affordmentioned, works using single-core architecture have a maximum throughput of 275.2 Gbps. Therefore, multi-core architectures using GPUs is a good choice to achieve high throughput. However, these GPU-based architectures always require huge power consumption. In the next section, we propose a multi-core AES architecture, which is implemented on ASIC technology to achieve ultra-high-throughput, but with power-efficiency and area-efficiency.

3. Proposed Hardware Architecture

3.1. Top Architecture

Although in ASIC technologies, several architectures of the AES reaching up to 100 Gbps throughput have been demonstrated in [10], [12], [13]. These architectures are not compatible with new data transfer standards such as IEEE P802.3bs 2017, which has a data transfer rate of up to 400 Gbps. So, we proposed the parallelization of multi-core AES. In each AES core, outer rounds pipeline technique is used to increase the operating frequency and minimize the latency.

The proposed multi-core AES architecture is described in Figure 2. This architecture consists of N single AES cores that work in parallel to speed up the encryption. So, each clock cycle encrypts 128×N bits of input data. Traditionally, each AES core has a Key Expansion block used to generate round keys for each round of AES. However, in our multi-core AES architecture we use only one Key Expansion block that is shared for all the AES cores in the architecture. This approach makes the proposed architecture more efficient in terms of power consumption and area cost while increasing the encryption throughput dramatically.

In each AES core, there are 10 CipherRounds, to speed up encryption, between rounds we insert registers to create the outer round pipeline architecture. Rounds of each AES single core consists of four major transformations: SubMatrix, ShiftMatrix, MixMatrix and AddRoundKey (as in Figure 3).

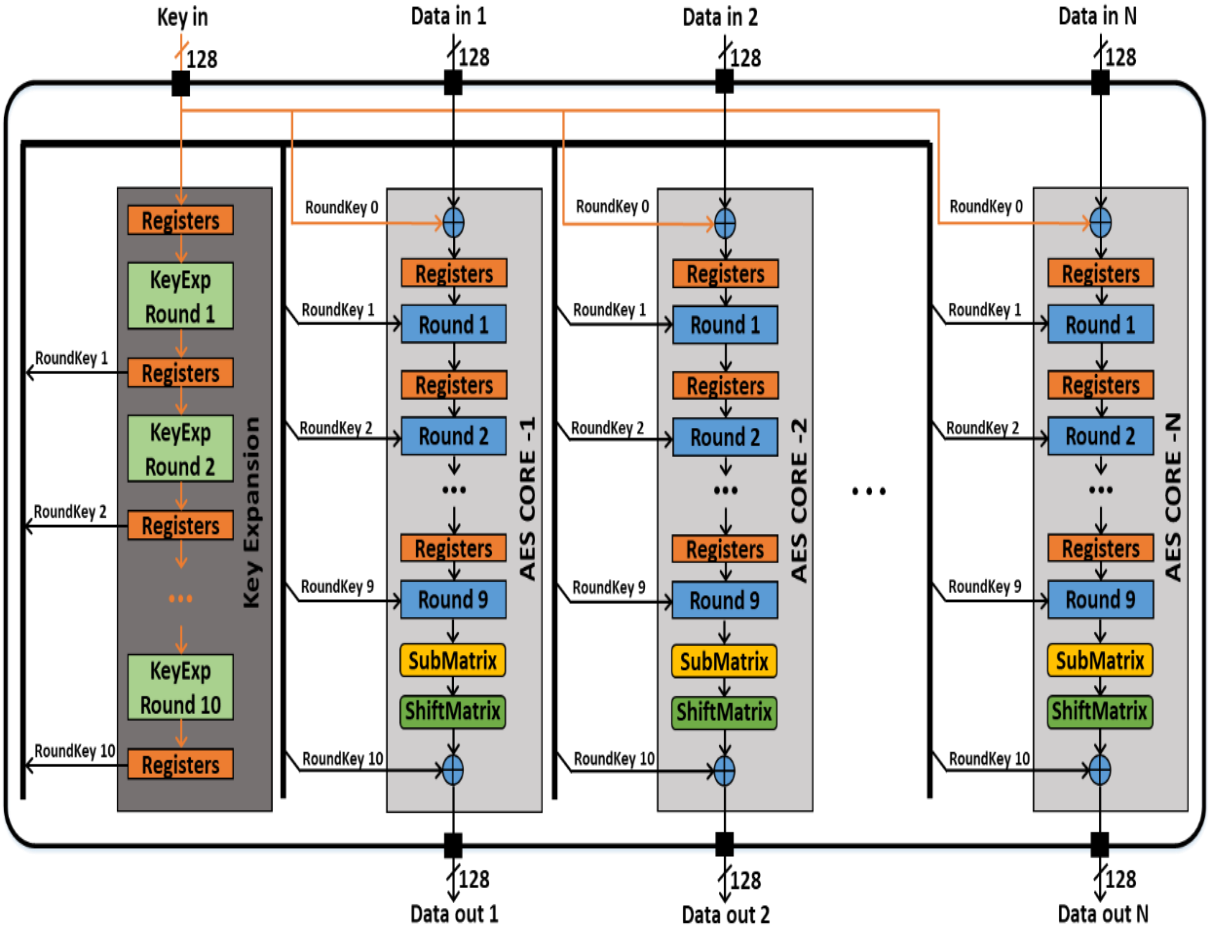

Figure 2. Architecture of Multi-core outer round pipelining and parallel AES-128.

3.2. Proposed CipherRound architecture

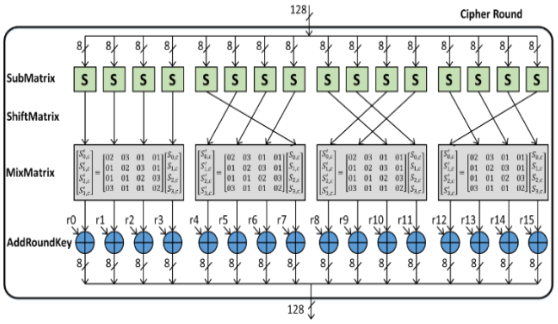

Figure 3. The micro architecture of CipherRound.

The micro architecture of CipherRound is shown in Figure 3. AES CipherRound employs consecutively four primary operations: SubMatrix, ShiftMatrix, MixMatrix, and AddRoundKey. The architectural details of these transformations are proposed in the following sub-sections.

3.3. Proposed SubMatrix and ShiftMatrix transformation

To speed up the encryption process, we apply parallelization techniques for each transformation. The input per round is 128 bits (16 bytes) assembled into a 4×4-byte matrix, so in the SubMatrix transform we use 16 S-Boxes, each S-Box is a 16×16-byte Look-Up Table. The architecture of SubMatrix is shown in Figure 4. Therefore, we use 16 S-Boxes for each transformation round.

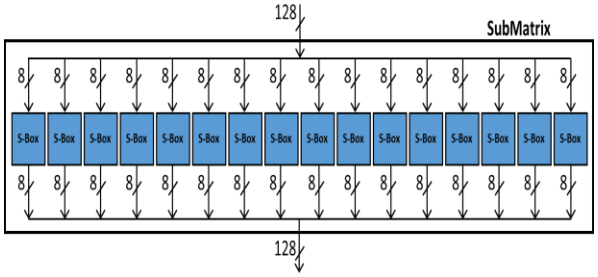

Figure 4. The micro architecture of Parallel S-Boxes in SubMatrix transformation.

In ShiftMatrix transformation, the 4×4-byte input data matrix is transformed as follows: The first row of the matrix is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed. The ShiftMatrix transformation is implemented through simple signal wiring.

3.4. Proposed MixMatrix transformation

MixMatrix transformation is composed of four MixColumns transformations. MixColumns transformation multiplies each column of the input matrix by matrix M (Eq. (3)). The multiplication of elements of $GF(2^8)$ in AES is accomplished by multiplying the corresponding polynomials modulo a fixed irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. Each element of this field can be treated as either an 8-bit string (in the binary or hexadecimal representation) or as a polynomial of degree seven or less, with coefficients in {0,1} (polynomial basis representation). The coefficients of a polynomial are equal to the respective bits of the binary representation [7].

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

Therefore:

$S'_{0,c} = (\{02\} \cdot S_{0,c}) \oplus (\{03\} \cdot S_{1,c}) \oplus S_{2,c} \oplus S_{3,c}$
$S'_{1,c} = S_{0,c} \oplus (\{02\} \cdot S_{1,c}) \oplus (\{03\} \cdot S_{2,c}) \oplus S_{3,c}$
$S'_{2,c} = S_{0,c} \oplus S_{1,c} \oplus (\{02\} \cdot S_{2,c}) \oplus (\{03\} \cdot S_{3,c})$
$S'_{3,c} = (\{03\} \cdot S_{0,c}) \oplus S_{1,c} \oplus S_{2,c} \oplus (\{02\} \cdot S_{3,c})$

Firstly, we calculate: $\{02\} \cdot S_{0.c}$

We have: {02} in hexadecimal is equivalent to {0000 0010} in binary, and to $\{02\} = 0.x^7 + 0.x^6 + 0.x^5 + 0.x^4 + 0.x^3 + 0.x^2 + 1.x + 0 = 1.x = x$ in the polynomial basis representation.

Similar, $S_{0,c} = b_7b_6b_5b_4b_3b_2b_1b_0$ (in binary) and $S_{0,c} = f(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$ (in polynomial).

So $\{02\} \cdot S_{0.c} = xf(x)$.

If $b_7 = 0$ then:

$$xf(x) = b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + +b_2x^3 + b_1x^2 + b_0x$$

If $b_7 = 0$ then:

$$xf(x) = b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + +b_2x^3 + b_1x^2 + b_0x \quad (4)$$

If $b_7 = 1$ then:

$$\begin{aligned} xf(x) &= (x^8 + b_6x^7 + b_5x^6 + b_4x^5 + +b_3x^4 + b_2x^3 + b_1x^2 + b_0x)\ mode\ m(x) \\ &= [(x^8 + x^4 + x^3 + x + 1) + (b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x) + \\ &\quad +(x^4 + x^3 + x + 1)]\ mode\ m(x) \\ &= [m(x) + (b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + +b_2x^3 + b_1x^2 + b_0x) + (x^4 + x^3 + \\ &\quad x + +1)]\ mode\ m(x) \\ &= (b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + +b_1x^2 + b_0x) + (x^4 + x^3 + x + 1) \end{aligned} \quad (5)$$

From (4) and (5), we have:

$$\begin{cases} b_7b_6b_5b_4b_3b_2b_1b_0 \times 00000010 = \\ \quad b_6b_5b_4b_3b_2b_1b_00 \quad if\ b_7 = 0 \\ b_6b_5b_4b_3b_2b_1b_00 \oplus 00011011 \quad if\ b_7 = 1 \end{cases} \quad (6)$$

So, we can write:

$\{02\} \cdot Byte = (b_6b_5b_4b_3b_2b_1b_0\ \&\ '0')\ xor\ "1B")$ when $b_7 = '1'\ else\ (b_6b_5b_4b_3b_2b_1b_0\ \&\ '0')$;

Secondly, we calculate: $\{03\} \cdot S_{0,c}$. Because of $0x03 = 0x02 \oplus 0x01$ so we have:

$$\begin{aligned} b_7b_6b_5b_4b_3b_2b_1b_0 \times 00000011 &= \\ b_7b_6b_5b_4b_3b_2b_1b_0 \times & \\ \times 00000010 \oplus b_7b_6b_5b_4b_3b_2b_1b_0 & \end{aligned} \quad (7)$$

In other words, $\{03\} \times \{Byte\} = \{02\} \times \{Byte\} \oplus \{Byte\}$.

From Eq. (6) and Eq. (7) we propose hardware architecture for MixColumn in Figure 5. In this figure, we used XOR gates with two inputs and MUX to get $\{02\} \times \{Byte\_In\}$ and $\{03\} \times \{Byte\_In\}$ and then calculate Byte_Out using XOR gates with 4 inputs.

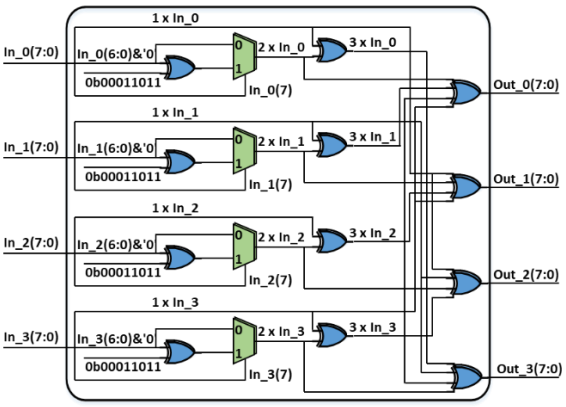

Figure 5. Architecture of MixColumn transformation.

Because of using a parallel decoding architecture, we used four MixColumn blocks together to create MixMatrix.

3.5. Proposed Key Expansion and AddRoundKey transformation

To share the same Key Expansion between different AES cores, we proposed an architecture of Key Expansion round to create RoundKeys for each AES round as depicted in Figure 6. This approach will help us to minimize the implementation overhead for the whole encryption architecture. Instead of using multiple Key Expansion blocks, only one Key Expansion has been used in our encryption architecture.

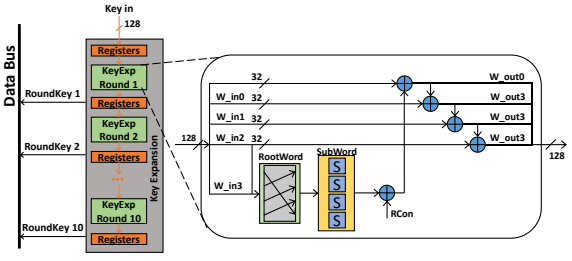

Figure 6. Architecture of Key Expansion round.

In this architecture, the 128-bit key of the previous round is divided into 4 words (each word is 32-bit) and brought to the input of the Key Expansion round. The Key Expansion will perform transforms:

- RootWord: Performs a one-byte circular left shift on a word.

- SubWord: Performs a byte substitution on each byte of its input word, using the S-box.

- Xored with the RCon register: The result of SubWord is XORed with a round constant Rcon[j]. With Rcon[j] = (RC[j], 0, 0, 0), where j is round index and the values of RC[j] in hexadecimal are:

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

The AddRoundKey transformation performs an operation on the State with one of the RoundKeys. The operation is a simple XOR between each byte of the State and each byte of the RoundKeys.

4. Results and Discussion

Firstly, we have modelled a single AES core at RTL in VHDL and then synthesized the design using 45nm CMOS technology with Synopsys Design Compiler. Table 2 summarizes single-core AES architecture implementation results using 45nm ASIC technology. The hardware synthesis results show that the throughput of single-core AES is 111.3 Gbps and KeyExpansion transformation (online key) uses 10.6% of the total area cost.

Table 2. Single core AES encryption implementation:

Technology	**45 nm**
Operating frequency (MHz)	833
Latency (ns)	12.6
Number of clocks per encryption	11
Throughput (Gbps)	111.3
Power consumption (mW)	56.3
Total area (mm^2)	0.11
Total area (kGate)	142
KeyExpansion area (mm^2)	0.012
KeyExpansion area (kGate)	15.1
KeyExpansion area/total area (%)	10.6

Therefore, a single pipelined AES is not able to achieve 400 Gbps or higher. This limitation forces the introduction of a multi-core construction. Because KeyExpansion transformation (online key) in single-core AES uses 10.6% of total area cost, to improve the area-efficiency we proposed an online shared-key multi-core AES architecture. We instantiated parallel AES cores sharing the same KeyExpansion transformation circuit. The AES plain modules work consequently using the same round keys. Figure 2 shows a schematic overview of the main components. The resulting multi-core design is thus able to process a $N \times 128$-bit block (where N is the number of AES cores) of plaintext at each clock cycle. Encryption throughput is defined as the number

of bits encrypted in a unit of time. Therefore, the throughput is calculated as follows:

$$Throughput(Gbps) = N \times 128(\text{bit}) \times F_{max}(Ghz) \quad (8)$$

The motivation for implementing multi-core AES architectures is to consider this architecture in terms of bandwidth, hardware resource efficiency and energy efficiency. The obtained results of hardware synthesis with the number of cores from 1 to 10 on 45nm CMOS technology are presented in Table 3. The comparison of the efficiency between different number of cores is presented Figure 7. In the single-core architecture, the throughput is 111.3 Gbps, but with 10 AES cores architecture on chip throughput up to 1 Tbps. Therefore, these architectures will meet the data encryption speed requirements for current communication standards and future. Our proposed 4-core architecture has a throughput of 433.7 Gbps that meets the 400 Gbps data transfer requirement of P802.3bs 2017. In Table 4 we suggest configuring the number of single AES cores on the chip to match in line with data rates of IEEE standards.

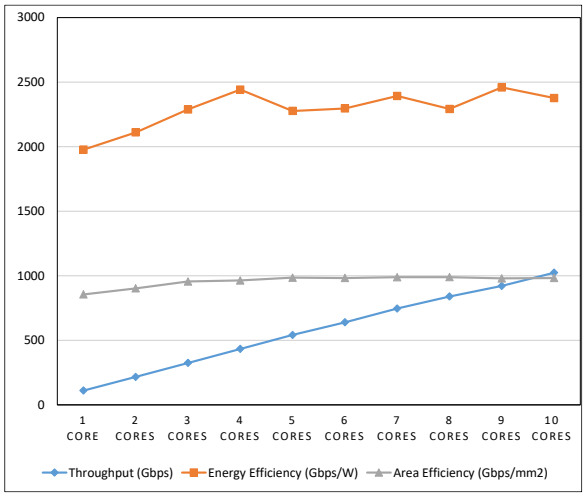

Figure 7. Multi-core comparisons.

Latency in AES encryption is defined by the number of cycles that each data sample takes to go through the encryption data-path before the encrypted output is generated. When there is only outer round pipelining (one stage pipeline per round), the latency is 11 cycles. In our design, we used 11 pipeline stages, so latency is 11 clock cycles:

$$latency(ns) = 11 \times T_{Clk} = 11/f_{max} \quad (9)$$

To increase the throughput, it is necessary to increase the number of single-cores AES on the chip. Because AES cores use the same KeyExpansion, it increases critical path delays. On the other hand, the clock tree is also bigger so the maximum operating frequency must be reduced. According to Eq. (9) when f_{max} decreases, the latency increases. The number of cores appropriate to the bandwidth will optimize power consumption, area and latency. In our design, the single-core architecture (N = 1) has a latency of 12.6 ns. With 4-AES cores on the chip (N = 4), the latency is 13 ns and when the number of cores increases to 10 (N = 10), the latency is 13.8 ns, lower than related works. In real-time applications, latency is an important factor. Delay in the encryption, decryption plus other types of delays can affect the quality of the service. Therefore, it is important to select the number of AES cores on the chip that are suitable for each application. In Table 4 we recommend multi-core AES configurations for specific applications.

Galois Counter Mode is a block cipher mode of operation that provides authenticated encryption via hashing over a binary Galois field of order 2^{128}, denoted GF(2^{128}). GCM, together with the block cipher AES, has been standardized for use in several network protocols including IPsec and MACsec. Our architecture is configurable to implement AES-GCM mode as shown in Figure 8.

Table 3. Implementation results of multi-core AES on 45nm CMOS technology:

Design	**F_{max}** *(MHz)*	**Area** *(mm^2)*	**Area** *(kGate)*	**Power** *(mW)*	**Throughput** *(Gbps)*	**Latency** *(ns)*	**Energy-efficiency** *(Gbps/W)*	**Area-efficiency** *(Gbps/mm^2)*	**Throughput/core** *(Gbps/core)*
1 core (N=1)	870	0.13	164.5	56.3	111.3	12.6	1977	856	111.3
2 cores (N=2)	847	0.24	303.6	102.7	216.8	13.0	2111	903	108.4
3 cores (N=3)	847	0.34	431.9	142.0	325.2	13.0	2289	956	108.4
4 cores (N=4)	847	0.45	561.1	177.6	433.7	13.0	2442	964	108.4
5 cores (N=5)	847	0.55	690.9	238.1	542.1	13.0	2277	986	108.4
6 cores (N=6)	833	0.65	815.2	278.7	639.7	13.2	2296	983	106.6
7 cores (N=7)	833	0.75	945.7	311.9	746.4	13.2	2393	989	106.6
8 cores (N=8)	820	0.85	1062.6	366.4	839.7	13.4	2292	990	105.0
9 cores (N=9)	800	0.94	1178.5	374.8	921.6	13.8	2459	980	102.4
10 cores (N=10)	800	1.04	1305.9	430.9	1024.0	13.8	2377	983	102.4

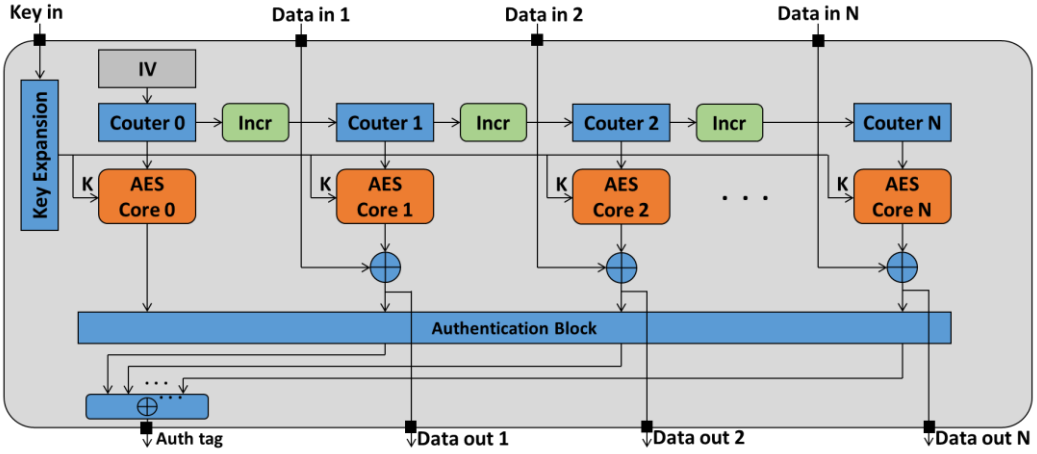

Figure 8. Proposal to use our architecture in AES-GCM mode.

Table 4. Recommend multi-core AES configurations for specific applications:

Design	**Throughput** *(Gbps)*	**IEEE Standard**
1 core (N=1)	111.3	802.1ae, 802.3ba, 802.3bj, 802.3bm (100 *Gbps*)
2 cores (N=2)	216.8	802.3 cd (50 to 200 *Gbps*)
3 cores (N=3)	325.2	-
4 cores (N=4)	433.7	802.3bs (200 to 400 *Gbps*)
5 cores (N=5)	542.1	For future
6 cores (N=6)	639.7	For future
7 cores (N=7)	746.4	For future
8 cores (N=8)	839.7	For future
9 cores (N=9)	921.6	For future
10 cores (N=10)	1024.0	For future

Power consumption and area are proportional to the number of cores on the chip. However, because the multi-core architecture shares the Key Expansion block, it is more energy-efficient and more efficient in using the area than the single-core architecture. With one core on the chip, the energy efficiency is 1977 Gbps/W and the area-efficiency is 956 Gbps/mm^2. With 10 cores on the chip, the energy efficiency is 2377 Gbps/W and the area-efficiency is 983 Gbps/mm^2. Therefore, the 10-core architecture is 20% more energy-efficient and 28% more area-efficient than the single-core architecture. On the other hand, compared to related works, our architecture is more efficient in terms of area and power consumption.

In some previous works that use GPUs to encrypt AES [23] (using Radeon HD 7970 GPU), the energy efficiency is 1.3 Gbps/W. At the same time, with 45nm CMOS technology, we achieve an energy efficiency of 2377 Gbps/W which is higher than 1828 times. However, in terms of throughput, our 10-core AES architecture reaches 1024 Gbps less than the works [22] (using Tesla V100 GPU), but higher than works using other GPU, FPGA and ASIC (Table 5).

Table 5. Throughput of multi-core AES encryption comparison:

Design	**Platform**	**Number of cores**	**Throughput (Gbps)**
[14] 2019	CMOS 65nm	9 cores AES CCM	13.54
[5] 2015	multiple FPGAs	20 core AES GCM	883
[15] 2010	FPGA Xilinx Virtex-5	4 cores AES GCM	119.3
[16] 2012	Intel® Xeon® X7560 Processors	32 cores	6.6
[17] 2017	NVIDIA GeForce GTX 1080 GPU	8 cores AES-ECB	279.86
[18] 2017	NVIDIA Tesla P100-PCIe	AES-ECB	605.9
[19] 2019	Tesla V100 GPU	AES-ECB	1380
[19] 2019	Tesla V100 GPU	AES- CTR	1470
[20] 2014	Radeon HD 7970	AES-ECB	205
Our work	CMOS 45nm	10 cores AES-ECB	1024

5. Conclusion

In this work, we have proposed a paralleled multi-core AES architecture which is able to provide ultra-high-throughput encryption flow. To minimize the design overhead in terms of hardware implementation area and power consumption, only one KeyExpansion block is shared between AES cores. The hardware performance results demonstrate that our architecture achieves an ultra-throughput of 1 Tbps with 10 AES cores on the chip.

Different AES cores use the same KeyExpansion unit, thus save area and power

consumption. With 10 AES cores, energy efficiency is 20% greater and the area-efficiency is 28% greater than those of a single-core architecture. The results of hardware synthesis are also compared with other works using FPGA, ASIC, GPUs,...

The outer pipelined and fully parallel architecture in each core reduces the critical path, thus increasing operating frequency and reducing latency. Our multi-core AES architecture has a low latency of 13.8 ns (with 10 AES cores). These results are lower than related works, so it is suitable for real-time applications. On the other hand, ultra-high-throughput of our design meets the data security requirements in new communication standards such as IEEE P802.3bm 2015, with providing data transmission at a bandwidth of 100 Gbps or IEEE P802.3bs 2017 has data transfer rates up to 400 Gbps.

Acknowledgment

This research is funded by the Ministry of Science and Technology of Vietnam under grant number KC.01.21/16-20 (ADEN4IOT).

References

[1] FIPS 197: Advanced Encryption Standard. National Institute of Standards and Technology, 2001.

[2] Cryptography and Network Security: Principles and Practice, Boston: Pearson, March 5, 2016.

[3] 802.3-2018 - IEEE Standard for Ethernet - IEEE Standard, https://ieeexplore.ieee.org/document/8457469, 2018, (accessed on: March 14th 2021)

[4] EEE Computer Society, IEEE Standard for Ethernet - Amendment 10: Media Access Control Parameters, Physical Layers, and Management Parameters for 200 Gb/s and 400 Gb/s Operation," IEEE Std 802.3bs-2017 (Amendment to IEEE 802.3-2015 as amended by IEEE's 802.3bw-2015, 802.3by-2016, 802.3bq-2016, 802.3bp-2016, 802.3br-2016, 802.3bn-2016, 802.3bz-2016, 802.3bu-2016, 802.3bv-2017, and IEEE 802.3-2015/Cor1-2017), 2017, pp. 1-372.

[5] B. Buhrow, K. Fristz, E. Daniel, A highly parallel AES-GCM core for authenticated encryption of 400 Gb/s network protocols, in 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig), 2015.

[6] A. Soltani, S. Sharifian, An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA, Microprocessors and Microsystems, Vol. 39, No. 7, 2015, pp. 480-493.

[7] P. Chodowiec, FPGA and ASIC implementations of AES, Cryptographic engineering, Springer, 2009, pp. 235-294.

[8] A. Hodjat, I. Verbauwhede, Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors, IEEE Transactions on Computers, Vol. 55, 2006, pp. 366-372.

[9] S. K. Mathew, F. Sheikh, M. Kounavis, S. Gueron and A. Agarwal, 53 Gbps Native GF(2^4)^2 Composite-Field AES-Encrypt/Decrypt Accelerator for Content-Protection in 45 nm High-Performance Microprocessors, IEEE Journal of Solid-State Circuits, Vol. 46, 2011, pp. 767-776.

[10] G. Sayilar, D. Chiou, Cryptoraptor: High throughput reconfigurable cryptographic processor, 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014.

[11] L. Ali, I. Aris, F. S. Hossain, N. Roy, Design of an ultra high speed AES processor for next generation IT security, Computers & Electrical Engineering, Vol. 37, No. 6, 2011, pp. 1160-1170.

[12] B. Erbagci, N. E. C. Akkaya, C. Teegarden, K. Mai, A 275 Gbps AES encryption accelerator using ROM-based S-boxes in 65nm, 2015 IEEE Custom Integrated Circuits Conference (CICC), 2015.

[13] P. K. Dong, T. X. Tu, N. K. Hung, A 45nm High-Throughput and Low Latency AES Encryption for Real-Time Applications, 2019 19th International Symposium on Communications and Information Technologies (ISCIT), 2019.

[14] A. A. Pammu, W. Ho, N. K. Z. Lwin, K. Chong, B. Gwee, A High Throughput and Secure Authentication-Encryption AES-CCM Algorithm on Asynchronous Multicore Processor, IEEE Transactions on Information Forensics and Security, Vol. 14, No. 4, 2019, pp. 1023-1036.

[15] L. Henzen, W. Fichtner, FPGA parallel-pipelined AES-GCM core for 100G Ethernet applications, in 2010 Proceedings of ESSCIRC, 2010.

[16] A. Barnes, R. Fernando, K. Mettananda, R. Ragel, Improving the throughput of the AES algorithm with multi-core processors, 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), 2012.

[17] A. Abdelrahman, M. Fouad, H. Dahshan, A. Mousa, High-Performance CUDA AES Implementation: A Quantitative Performance Analysis Approach, Computing Conference 2017, London, UK, 2017.

[18] N. Nishikawa, H. Amano, K. Iwai, Implementation of Bitsliced AES Encryption on CUDA-Enabled GPU, in Network and System Security: 11th International Conference, 2017.

[19] O. Hajihassani, S. Monfared, S. H. Khasteh, S. Gorgin, Fast AES Implementation: A High-throughput Bitsliced Approach, IEEE Transactions on Parallel and Distributed Systems, 2019.

[20] N. Nishikawa, K. Iwai, H. Tanaka, T. Kurokawa, Throughput and Power Efficiency Evaluation of Block Ciphers on Kepler and GCN GPUs Using Micro-Benchmark Analysis, IEICE Transactions on Information and Systems, Vol. E97.D, No. 6, 2014, pp. 1506-1515.

[21] M. S. Al-Bahri, A. J. AiShebani, K. Gupta, O. K. AlAwaisi, AES Parallel Implementation on a Homogeneous Multi-Core Microcontroller, 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018.

[22] H. Tazeem, M. Farid, A. Mahmood, Improving security surveillance by hidden cameras, Multimedia Tools and Applications, Vol. 76, No. 2, 2017, pp. 2713-2732.

[23] M. Wang, C. Su, C. Horng, C. Wu and C. Huang, Single- and Multi-core Configurable AES Architectures for Flexible Security, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 4, 2010, pp. 541-552.

[24] K. Rahimunnisa, P. Karthigaikumar, N. Christy, S. Kumar, J. Jayakumar, PSP: Parallel sub-pipelined architecture for high throughput AES on FPGA and ASIC, Central European Journal of Computer Science, Vol. 3, No. 4, 2013, pp. 173-186.

[25] A. Hodjat, I. Verbauwhede, A 21.54 Gbits/s fully pipelined AES processor on FPGA, 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004.

[26] J. Ma, X. Chen, R. Xu, J. Shi, Implementation and Evaluation of Different Parallel Designs of AES Using CUDA, 2017 IEEE Second International Conference on Data Science in Cyberspac, 2017.

[27] R. Lim, L. Petzold, Ç. Koç, Bitsliced High-Performance AES-ECB on GPUs, The New Codebreakers: Essays Dedicated to David Kahn on the Occasion of His 85th Birthday, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 125-133.

[28] https://developer.nvidia.com/cuda-zone, (accessed on: March 14th 2021).

[29] Q. Liu, Z. Xu, Y. Yuan, High Throughput and Secure Advanced Encryption Standard on Field Programmable Gate Array with Fine Pipelining and Enhanced Key Expansion, IET Computers Digital Techniques, Vol. 9, No. 3, 2015, pp. 175-184.