Efficient Architecture for a High Performance Authenticated Encryption Algorithm on Reconfigurable Computing

Abiy Tadesse Abebe[1(✉)], Yalemzewd Negash Shiferaw[1], and P. G. V. Suresh Kumar[2]

[1] School of Electrical and Computer Engineering, Addis Ababa Institute of Technology, AAU, Addis Ababa, Ethiopia
{abiy.tadesse,yalemzewdn}@aait.edu.et
[2] Ambo University, Ambo, Ethiopia

Abstract. High performance authenticated encryption algorithms are indispensable and preferable for securing the contemporary high speed wireless networks as they can perform their tasks without affecting overall performance of the network, and can provide data confidentiality, data integrity, and authentication cryptographic services simultaneously. Most of existing FPGA based architectures that have been proposed to enhance performance of such algorithms considered generic FPGA fabrics for implementations. Implementing complex algorithms using only traditional FPGA logic requires large amount of such resources that in turn can affect performance. In this work, an efficient architecture for AEGIS-128 authenticated encryption algorithm is proposed using both FPGAs' embedded hard-cores such as digital signal processing slices and block random access memories that have not been fully exploited for such applications with balanced amount of generic logic. The aim is to reduce performance bottlenecks and enhance performance of AEGIS-128. The implementation results show that the proposed architecture outperforms existing similar approaches found in the literature in terms of throughput, and utilization of reduced amount of resources.

Keywords: AEGIS-128 · AES-128 · BRAMs · DSP slices · Cryptosystem · Embedded hard-cores · FPGA

1 Introduction

High performance cryptographic algorithms are useful to secure the contemporary high speed wireless networks since they can perform their security tasks without affecting the overall performance of the network [1]. Authenticated encryption algorithms are preferable for the contemporary information security since a single algorithm can provide multiple cryptographic security services

M. L. Berihun (Ed.): ICAST 2021, LNICST 411, pp. 575–585, 2022.
https://doi.org/10.1007/978-3-030-93709-6_39

including data confidentiality, data integrity, and data origin authentication, simultaneously [1]. These algorithms can be classified as high performance and lightweight Authenticated Encryption with Associated Data (AEAD). The former has been designed targeting high speed applications. Whereas, the latter is designed mainly by considering the resource and performance limitations of constrained devices. High speed authenticated encryption algorithms are useful to secure high performance applications since they can offer the required high throughput if they are efficiently implemented. There are several Advanced Encryption Standard (AES) based high performance authenticated encryption algorithms [1,2]; and, many researchers have proposed different FPGA based architectures to enhance performance of such high speed authenticated encryption algorithms [3]. However, most of them considered the generic FPGA fabrics to implement their proposed architectures. But, in addition to the traditional FPGA logic elements, modern FPGAs have also incorporated embedded hard-cores that have not been fully exploited for implementation of such algorithms. Implementation of complex algorithms using only the common FPGA logic elements have some limitations such as requirement of large amount of such resources that in turn can impact performance.

The main contributions of this research work are described as follows:

- Efficient architecture for AEGIS-128 authenticated encryption algorithm tailored to the security of high performance applications is proposed synthesized, optimized, and implemented on FPGA achieving high throughput results.
- Improving throughput with reasonable hardware resource utilization are the optimization targets for the proposed architecture. Therefore, a hybrid optimization technique including as parallel-pipelining approach is used to meet the intended optimization targets.
- The potential use of modern FPGA resources such as embedded hard-cores including Digital Signal Processing (DSP) slices and Block Random Access Memories (BRAMs) are exploited to implement the proposed architectures with balanced utilization of the traditional FPGA fabrics. This approach enabled to enhance performance of AEGIS-128 algorithm in terms of throughput and resource utilization.

The paper is organized as follows:
Summary of related works are discussed in Sect. 2. Section 3 briefly describes the core of AEGIS algorithm, the Advanced Encryption Standard (AES) and its FPGA based implementation techniques. Efficient AES architecture for FPGA implementation is presented in Sect. 4. Section 5 briefly highlights the state update procedures of AEGIS-128 algorithm based on AES. Section 6 presents the proposed efficient architecture for AEGIS-128. Implementation approaches for the proposed AEGIS-128 architecture on FPGA and result comparisons are presented in Sect. 7. Finally, Sect. 8 concludes the paper.

2 Summary of Some Existing Related Works

There are some proposed architectures for AEGIS algorithm including both high speed and reduced area implementations on FPGA and ASIC platforms.

In [3], two different FPGA based AEGIS-128L architectures were proposed, one for reduced area and the other for high speed using looping and pipelining methods, respectively. In case of pipelining method, they used BRAMs for storing and loading of keys. However, they used a straightforward pipelining technique, but no any other technique was employed. Moreover, they didn't mention about whether they used full pipelining, inner pipelining or both. ASIC based implementation results of various AEAD algorithms were reported including AEGIS algorithm in [4] using HLS tool, but no specific implementation technique was described. In [5], compact ASIC based AEGIS architecture was proposed for secure FPGA reconfiguration. Since a single AES round consists 16 SubBytes(s-box), a ShiftRows, and four MixColumns operations, to reduce the area, they used only four s-boxes; and then, using only one mix-column step. As a consequence, they achieved $1/4^{th}$ of the area consumed if, otherwise, a full round of AES is used. However, their architecture was not evaluated based on FPGA. In the works presented in [6] and [7], full outer and inner pipelining techniques were employed to achieve high throughput.

3 The Core of AEGIS-128 Algorithm: AES

The core of AEGIS-128 authenticated encryption algorithm [1] is AES [8]. AES is a 128 bit block cipher algorithm with three different key sizes, 128 bit, 192 bit and 256 bit with 10, 12 and 14 round operations, respectively. The AES encryption process performs four basic operations: SubBytes (byte substitution), ShftRows (rows shifting), MixColumns (column mixing) and AddRoundKey (adding round keys) to provide data confidentiality. In the last round, there is no MixColumns transformation. Before starting execution of these four basic operations, the 128 bit block data (plaintext) is arranged in to a 4x4 matrix which is known as a state. Each cell in a state contains elements with 8 bits size and are named as state elements. The state is XORed with the initial round key before the round operations are started. The initialization process is done by adding the first round key (128 bits) with 128 bits plaintext. Then, all the four operations are performed consecutively on the resulting state outputs (after each round step), until the specified round number is reached. The final result of the encryption process provides the ciphertext. For each round operation a separate key is used which is generated by using a key generator function [8].

The decryption process generally involves the inverse steps of the encryption rounds: InverseSubBytes, InverseShiftRows, AddRoundKey (which is the inverse of itself), and InverseMixColumns. Details about the construction of AES can be found in [8], and [9].

3.1 Optimization Techniques for FPGA Based AES Implementation

The standard AES algorithm can be efficiently implemented on FPGA targeting small area or high speed optimization metrics, or else the trade-off between them. Small area optimization refers to utilization of small amount of FPGA resources

for compact implementation [10] results specially tailored to constrained environments. An example of small area architecture of AES implementation was presented in [11]. This approach used a single round and iteratively processed it ten times for AES-128. The throughput can be obtained by computing the product of the maximum frequency achieved and the data block size (128), and dividing the result by ten (number of rounds) [11] as shown in Eq. 1:

$$Throughput(Mbps) = \frac{F_{max}(MHz) \times 128}{10} \tag{1}$$

It is possible to improve the throughput outcome with cost of increased area by using loop unrolling or pipelining architectures. In pipelining architecture [12] registers are placed at each round of AES to construct the pipeline. It is possible to determine the depth of the pipeline that can limit the number of the data blocks to be processed concurrently. If fully pipelining is required so as to achieve higher throughput, the total number of rounds of AES is taken as the depth of the pipeline [12,13]. Generally, pipelining increases the encryption speed by processing multiple blocks of data simultaneously. Similar to pipelining, the sub-pipelining architecture is formed by further putting registers inside of each round function of AES. If each round unit has n stages with equal delay, then a K-round sub-pipelined architecture can achieve approximately n times the speed of a K-round pipelined architecture with some increase of area caused by additional registers and control logic. In loop unrolling architecture, registers inserted for pipelining are removed and all rounds are independently implemented in hardware [12]. In this case, multiple AES rounds are processed in the same clock cycle. The delay of each round is the same which depends on the combinational logic used. Unrolled architectures have a large number of rounds that are independently implemented in hardware, and can increase hardware complexity.

4 The Proposed Architecture for AES on Reconfigurable Computing

Combining existing approaches, a hybrid technique is proposed for FPGA based efficient AES implementation. As shown in Fig. 1, the proposed hybrid technique combines: two AES-128 cores working in parallel for high speed, full outer pipelining that pipeline all the AES rounds, inner parallel round functions with partial sub-pipelining that enable selected groups of round operations to be executed in parallel-pipelining mode. The pre-stored BRAMs are used to provide the plaintext (P_t) in parallel mode. The initial round is first performed by XORing the P_t with the initial round key in both AES cores in parallel. Then, the intermediate rounds are performed similarly by XORing the processed state output of each round with the AddRoundKey of their respective round steps in both AES-cores in parallel. After the processing of intermediate rounds are completed, the last rounds are executed in parallel producing the ciphertext (C_t) for

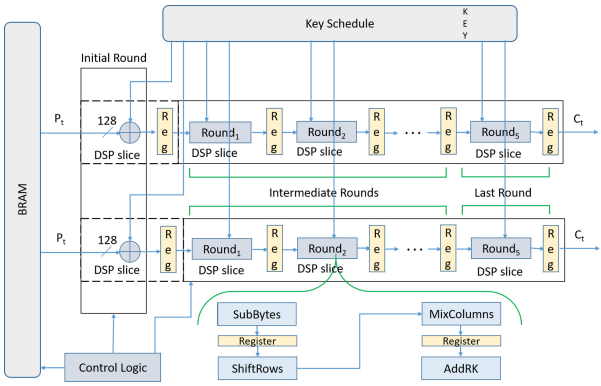

Fig. 1. Structure of the combined AES architecture

both AES-cores. The inner rounds in both paths are partially pipelined and processed in parallel in groups of two in such a way that SubBytes and ShiftRows are processed in parallel with MixColumns and AddRoundKey as shown in Fig. 1, except for the last round that doesn't include MixColumn. The pipelined registers are used to speed up the process. The same round keys are used for both cores. All these operations are performed until all the input message is processed.

The key schedule produces the required keys and stores them in BRAMs, and they are used during execution of the two AES-cores. Key-expansion in AES-128 is the process of generating all Round Keys from the original input key. Ten Round keys are generated with 16 bytes size for the 128 bits key size.

The proposed method differs from existing related architectures in that it combines various techniques to provide an AES-128 architecture based on hybrid approach and implements it using DSP slices and BRAM hard-cores in addition to balanced utilization of the traditional FPGA fabrics, instead of using only one resource type, and excluding the other. This increases flexibility of implementation while exploiting the modern FPGA resources for suitable parts of the algorithm under consideration. The combined effect of the proposed architecture is intended to produce a balanced trade-off between speed and area giving more emphasis to speed.

4.1 Implementation Approaches

For implementation of this architecture, both FPGA generic fabrics and dedicated hard-cores including DSP48E1 and BRAMs are used in order to balance hardware resource utilization and enhance throughput efficiency. The proposed architecture is implemented on Xilinx Virtex-5, Virtex-7, and Zynq FPGA devices. Xilinx Vivado 2018.3 High-level synthesis tool is used to test the proposed architecture's functionality. Figure 2 shows functional verification for encryption part of AES-128. Xilinx Vivado HLS is a modern EDA tool and is state-of-the-art technology which enables to specify the design in software. It

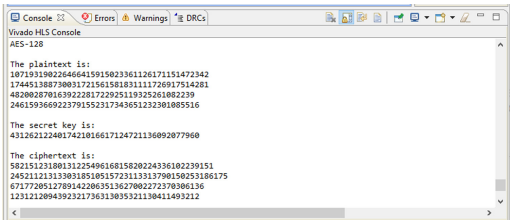

Fig. 2. Functional test of AES-128 using Vivado HLS

can also synthesizes it and produces RTL of the design, and provides flexibility to optimize the design using several optimization directives so as to achieve the desired optimization metrics. The synthesized VHDL code is implemented targeting the specified devices. For older versions of Xilinx FPGA devices that are not supported by Vivado HLS tool, Xilinx ISE 14.5 is used to implement and analyze the performance of the proposed architecture.

Control logic is used to synchronize the parallel operations. And, pipeline registers found in the DSP slices are used for pipelining. Since the proposed architecture is in turn intended to optimize AEGIS-128 algorithm, only encryption part is considered.

Since binary addition operations in hardware are performed using Exclusive-or (XOR) operation, DSP slices are used to perform the XOR operations. The S-box values and constants are pre-stored in BRAMs. Therefore, the BRAMs are read whenever the data and keys are needed for processing the AES states. Cascaded DSP slices are used to perform 128 bit operations.

5 Brief Description of AEGIS-128 Algorithm

AEGIS-128 algorithm [1] is an Advanced Encryption Standard (AES) [8] based algorithm which uses the AES round functions excluding the last round. AEGIS-128 algorithm uses 128 bits key and 128 bits Initialization Vector (IV) to perform authenticated encryption and authenticated decryption processes. It can process less than 2^{64} lengths of plaintext and the associated data. The recommended length of the tag to be used for authentication is 128 bits; though, lesser lengths of tags can also be used [1].

The algorithm uses the AES round functions to update the 640 bits (80 bytes) of state, S_i, with 128 bits (16 bytes) of data blocks, m_i, using its state update function such that: $S_{(i+1)} = StateUpdate128(S_i, m_i)$ as shown in Fig. 3. In Fig. 3, R indicates the AES encryption round function, and w is a temporary 16-byte word [1]. The structure represented in Fig. 3 can also be expressed as follows [1]:

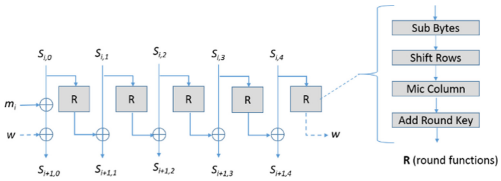

Fig. 3. The state update function of AEGIS-128

$$S_{i+1,0} = AESRound(S_{i,4}, S_{i,0} \oplus m_i);$$

$$S_{i+1,1} = AESRound(S_{i,0}, S_{i,1});$$

$$S_{i+1,2} = AESRound(S_{i,1}, S_{i,2});$$

$$S_{i+1,3} = AESRound(S_{i,2}, S_{i,3});$$

$$S_{i+1,4} = AESRound(S_{i,3}, S_{i,4});$$

AEGIS-128 performs different execution steps including initialization, process of the associated data, encryption, and finalization. The decryption process requires the exact values of key size, IV size, and tag size to perform the verification and decryption tasks [1]. Details about AEGIS AEAD algorithm can be found in [1].

6 Efficient Architecture for AEGIS-128 Algorithm

Since AEGIS-128 algorithm [1] composes advanced encryption standard (AES) algorithm [8] as its central component, optimized architecture of AES in turn optimizes the AEGIS algorithm too. Therefore, the architecture proposed for optimization of AES in Sect. 4 is applied here for optimizing the AEGIS-128 algorithm with some rearrangements based on the specific requirements of AEGIS-128 algorithm. The AES algorithm implemented in Sect. 4 is AES-128. It is selected to fit AEGIS-128 architecture. Though AES-128 consists 10 rounds, AEGIS-128 algorithm uses only five rounds for state update function. Thus, the architecture proposed in Sect. 4 for AES-128 algorithm is modified to use only five rounds; and, the last round is removed as specified in AEGIS algorithm design [1]. Then, two AES-128 cores with only five rounds each, are made to be processed in parallel while maintaining the internal construction of the proposed AES architecture specified in Sect. 4. The structure of the proposed AEGIS-128 architecture is shown in Fig. 4. In addition to the optimized AES architecture, various pipelining stages are constructed to increase the AEGS-128 performance.

As shown in Fig. 4, the two AES-128 cores are executed in parallel each taking 128 bit plaintext independently. In addition, the five AES rounds are fully pipelined in each AES core. Moreover, the round functions in each round are

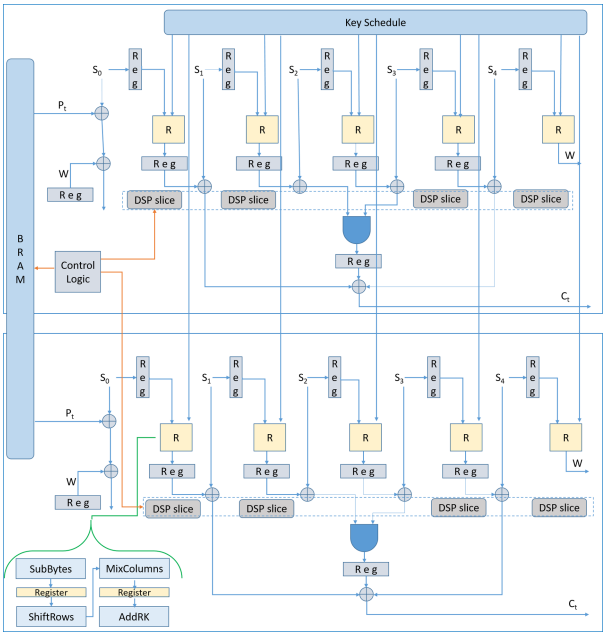

Fig. 4. Structure of the proposed AEGIS-128 architecture

also partially pipelined and are processed in parallel in groups of two similar to the AES architecture presented in Sect. 4.

In Fig. 4, the blocks labeled with R consist the four round operations: $SubBytes$, $ShiftRows$, $MixColumns$, and $AddRoundKey$. Whereas, the blocks indicated by Reg represent registers inserted between each round and inter-round for pipelining. The BRAM contains the required inputs values. The Key Schedule provides the required keys for each round operation and stores them in BRAMs to be executed as needed.

In this work, all the five rounds of the AES algorithm are made to process the state of AEGIS-128 concurrently in one clock cycle to update the state and achieve high throughput. The general throughput can be calculated as shown by Eq. 2:

$$Throughput(TP) = Max.Frequency \times 128 \tag{2}$$

The two parallel AES cores, the outer and inner pipelining, and the parallel and pipelined AEGIS-128 constructions, all add to fast processing and throughput increment by four times the throughput that can be achieved based on Eq. 2.

7 Implementation Approaches and Result Comparisons

The proposed AEGIS-128 architecture has been implemented targeting Xilinx Virtex-5 and Virtex-7 FPGA platforms. Xilinx Vivado HLS 2018.3 is used to perform functionality test for both authenticated encryption and authenticated decryption. Figure 5 and Fig. 6 show the functional test results for authenticated encryption and authenticated decryption, respectively. Then, synthesis and implementation of AEGIS-128 have been performed on Virtex-7. The obtained RTL output has also been synthesized and implemented using Xilinx ISE 14.5 tool specifically for Virtex-5 device since it is not supported by Vivado HLS tool directly. As shown in Table 1, for implementation of the proposed architecture on Virtex-5 FPGA device, higher throughput (60.949 Gbps) is achieved compared to existing research results found in the literature. Since the target of optimization for AEGIS-128 is to achieve high throughput, it is achieved with balanced resource utilization.

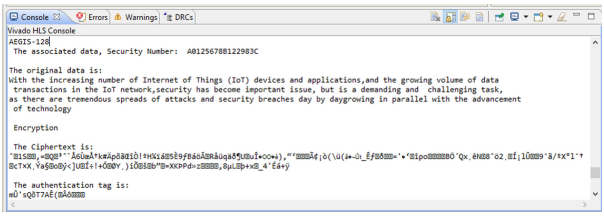

Fig. 5. Functional test for AEGIS-128 for authenticated encryption

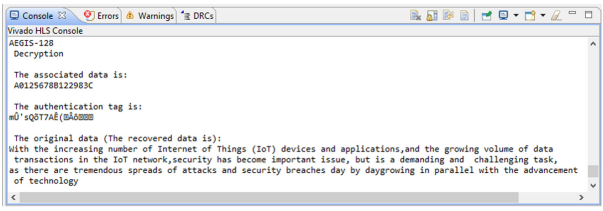

Fig. 6. Functional test for AEGIS-128 for authenticated decryption

Similarly, for implementation of the proposed method on Xilinx Virtex-7 FPGA device, both utilization of smaller slices and higher throughput result are achieved compared to existing research outcomes found in the literature.

The result comparisons are also analyzed graphically as shown in Fig. 7 and Fig. 8 for resource utilization and throughput performance on both Virtex-5 and Virtex-7 platforms, respectively.

Table 1. Comparison the proposed AEGIS-128 implementation results against existing outcomes

Methods	Device	Slices	LUTs	BRAMs	DSPs	Freq. (MHz)	Throughput (Mbps)
[7] pipelined	Virtex-5	5478	–	4	–	324.6	41.55 Gbps
[14] LUT based	Virtex-5	1391	–	–	–	156.5	20.03 Gbps
[6] Pipelined	Virtex-5	5586	–	6	–	348.7	44.6 Gbps
This work: hybrid	Virtex-5	1282	–	8	36	476.168	60.949 Gbps
[3]							
Basic	Virtex-7	9646	–	–	–	–	68121
Looping	Virtex-7	7726	–	–	–	–	64497
Pipelined	Virtex-7	10610	–	–	–	–	88564
[7] Pipelined	Virtex-7	9306	–	–	–	–	89354
This work: hybrid	Virtex-7	1375	–	8	36	–	99412

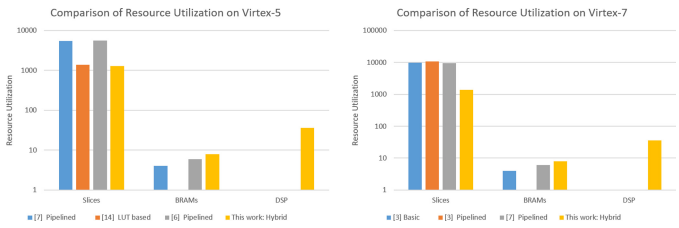

Fig. 7. Comparison of resource utilization

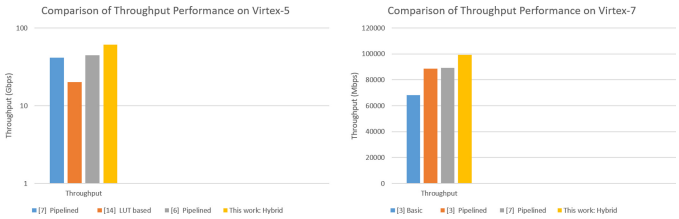

Fig. 8. Comparison of throughput performance

8 Conclusions

Efficient architecture for high performance authenticated encryption algorithm is proposed and implemented on FPGA considering requirements of high speed applications. Improving throughput with balanced hardware resource utilization is the optimization target of this work. Hybrid optimization techniques are employed and FPGAs' hard-cores together with reasonable amount of generic FPGA logic elements are used to enhance performance. Therefore, state-of-the-art optimization tools have been used and better performances are achieved. Therefore, both enhanced throughput and balanced resource utilization optimization targets have been met; and, the required performance metrics have been achieved based on the proposed architectures.

References

1. Wu, H., Preneel, B.: AEGIS: a fast authenticated encryption algorithm (v1. 1). In: Submission to CAESAR (2016)
2. McGrew, D., Viega, J.: The Galois/counter mode of operation (GCM). In: Submission to NIST (2005)
3. Katsaiti, M., Sklavos, N.: Implementation efficiency and alternations, on CAESAR finalists: AEGIS approach. In: 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp. 661–665. IEEE (2018)
4. Kumar, S., Haj-Yihia, J., Khairallah, M., Chattopadhyay, A.: A comprehensive performance analysis of hardware implementations of CAESAR candidates. IACR Cryptol. ePrint Arch., 1261 (2017)
5. Abdellatif, K. M., Chotin-Avot, R., Mehrez, H.: AEGIS-based efficient solution for secure reconfiguration of FPGAs. In: Proceedings of the Third Workshop on Cryptography and Security in Computing Systems, pp. 37–40. ACM (2016)
6. Abebe, A.T., Shiferaw, Y.N., Kumar, P.G.V.S.: Efficient reconfigurable integrated cryptosystems for cybersecurity protection. In: Shandilya, S.K., Wagner, N., Nagar, A.K. (eds.) Advances in Cyber Security Analytics and Decision Systems. EICC, pp. 57–77. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-19353-9_4
7. Tadesse Abebe, A., Negash Shiferaw, Y., Kumar, P.G.V.S.: reconfigurable integrated cryptosystem for secure data exchanges between fog computing and cloud computing platforms. In: Habtu, N.G., Ayele, D.W., Fanta, S.W., Admasu, B.T., Bitew, M.A. (eds.) ICAST 2019. LNICST, vol. 308, pp. 492–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43690-2_35
8. Daemen, J., Rijmen, V.: The design of Rijndael: AES - the advanced encryption standard. Information Security and Cryptography, Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04722-4
9. PUB, NIST FIPS. 197. Specification for the advanced encryption standard (AES) (2001). Accessed 26 Nov 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
10. Rajasekar, P., Mangalam, H.: Design and implementation of power and area optimized AES architecture on FPGA for IoT application, Circuit World (2020)
11. Standaert, F.-X., Rouvroy, G., Quisquater, J.-J., Legat, J.-D.: Efficient implementation of Rijndael encryption in reconfigurable hardware: improvements and design tradeoffs. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 334–350. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45238-6_27
12. Tadesse, A., Kumar, P.S.: Effective implementations techniques for FPGA based AES algorithm. In: 2016 KICS Korea and Ethiopia ICT International Conference (2016)
13. Zhang, X., Parhi, K.K.: High-speed VLSI architectures for the AES algorithm. IEEE Trans. Very Large Scale Integr. (VLSI) Syst **12**(9), 957–967 (2004)
14. Abdellatif, K.M., Chotin-Avot, R., Mehrez, H.: AES-GCM and AEGIS: efficient and high speed hardware implementations. J. Signal Process. Syst. **88**(1), 1–12 (2017)