ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

Efficient Advanced Encryption Standard Implementation with Enhanced Security Features

B.Satyanarayana, Dr M.Srinivasan

Department of ECE, Sri Satya Sai University of Technology & Medical Science, Sehore, Bhopal, Madhya Pradesh, India.

Department of ECE, Sri Satya Sai University of Technology & Medical Science, Sehore, Bhopal, Madhya Pradesh, India.

ABSTRACT: Data transferred in an electronic way is vulnerable to attacks. With an aim to protect data for secure communication, a new Hybrid non pipelined Advanced Encryption Standard (AES) algorithm based on traditional AES algorithm with enhanced security features is proposed in this work. Abysmal analysis of the AES algorithm implies that the security of AES lies in the S-box operations. This paper presents a new approach for generating S-box values (modified S-box) and initial key required for encryption/encryption (improved key generation) using PN Sequence Generator. The AES algorithm with proposed modifications shows significant improvement in the encryption quality as compared to traditional AES algorithm. The traditional AES algorithm equipped with proposed novel modified S-box technique and improved key generation technique gives an avalanche effect of 60% making it invulnerable to attacks. The proposed design is synthesized on various Field Programmable Gate Array (FPGA) devices and compared to the existing designs resulting in significant improvement in throughput. The proposed design is implemented on Spartan6 FPGA device.

I. INTRODUCTION

With the expansion of data communications and its applications, there is a greater demand for increasing security systems and devices to guard individual information sent over the transmission channel. One of the most important techniques for securing the information is data encryption. Two kinds of cryptographic techniques viz. symmetric and asymmetric cryptosystems have already been created and are widely used (Mohammed et al., 2011). Symmetric cryptography techniques, such as the DES, Triple DES and AES, utilize a key that is identical to the transmitter as well as a receiver, for encrypting and decrypting the data transferred. The Asymmetric cryptography techniques, such as the RSA, ECC and DSA make use of different keys for encrypting and decrypting the data transferred (Ebrahim et al., 2014). For securing large amount of data symmetric cryptography is much more appropriate. The AES algorithm, identified by the NIST of the United States of America is approved to displace DES (Fips-197, 2001). Analysis of cipher strength is an essential part of security assessment of any corporate or academic entity. Cipher analysts have indicated that, out of 10 rounds of AES, about 8 rounds can be brute forced successfully on today's modern day hardware systems. The remaining 2 rounds cannot be broken in sufficient duration so as to make the attack on the system meaningful to the attacker (Bogdanov et al., 2011). Due to the current trend of increase in computational power, it may not be long that the entire AES cipher would be de-ciphered under a given duration, and hence compromise the system under test. Consequently extensive research is being currently carried out to identify techniques to further secure AES algorithm. In this work, a PN Sequence Generator is used generation of S-box values and the initial key required for Encryption/Decryption. A PN Sequence Generator gives distinct values which satisfies the criterion for S-box values. These techniques result in enhancing the security of the AES algorithm as the feedback taps and seed value of the PN Sequence Generator are not known to an attacker and will make the algorithm invulnerable to brute force attack. The key contributions of this work are as follows: A new approach for generation of S-box values using PN

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

Sequence Generator is presented. A PN sequence generator generates a distinct sequence of random numbers based on the initial seed value and feedback taps. This property of a PN Sequence Generator is used for generating dynamic S-box which enhances the strength of the cryptosystem. This work presents a PN Sequence Generator based design of a Key Generation Block for generating initial key internally. A new secure initial key generation technique eliminates the need to apply the key externally and hence strengthens the modified AES algorithm against attack as compared to traditional AES algorithm. The techniques proposed in this paper are synchronized at both encryption and decryption ends for a truly secure AES algorithm. The AES algorithm with modified S-box values and improved key generation technique is tested on Strict Avalanche Criterion for key sensitivity and an avalanche effect of 60% is achieved. Also the proposed design is optimized for speed and area and compared with existing FPGA implementations. The implementation of AES algorithm with modified S-box values using Spartan6 FPGA device achieves a throughput of 3.039 Gbps with latency of 10 clock cycles. Organization of this paper is as follows: Section 2 presents the related work reported by various authors in the literature. Section 3 presents the description of AES algorithm. In Section 4, we propose the techniques and explain in details the need and advantages of generation of S-box values and initial key using PN Sequence Generator for enhancing the security of AES algorithm. Section 5 presents simulation results and comparison of the results with previous works reported in the literature. The conclusion of this work is stated in Section 6.

II. LITERATURE SURVEY

AES implementations are categorized into software and hardware implementations. Hardware implementation offers faster speed, more security and consumes less power and thus is an attractive choice as compared to software implementation. Implementation of algorithms on hardware can be achieved using either ASIC or FPGA devices. The authors presented ASIC implementation of the cryptographic algorithm. ASIC is an integrated circuit designed for specific application and lacks flexibility. Moreover, increased Non Recurring Engineering cost makes small volume production unaffordable. As compared to ASIC, an FPGA device is reconfigurable, efficient, offers more flexibility and requires less time to market and hence FPGA is a popular choice for hardware implementation. FPGAs also achieve much better performance than multi-core CPUs for mathematical computations. Significant amount of research has been done on hardware implementation of AES algorithm using FPGA. These hardware implementations aims at achieving increased throughput increased operating frequency, decreased latency, lesser area and lower power dissipation. The existing AES algorithm implementations are based on iterative structure and loop unrolled structures to minimize area and maximize operating frequency or obtain a trade-off between area and operating frequency. In iterative structures used for AES implementation only one cipher round is unrolled and data is iteratively looped until the entire encryption/decryption is completed. This results in reduced area utilization. In loop unrolled structures, all the cipher rounds are unrolled and data is looped through the cipher rounds sequentially until the entire encryption/decryption is completed (Ali et al., 2011; Standaert et al., 2003; Zhang and Parhi, 2004). The loop unrolled structure implementation, results in increased speed and increased area utilization. The AES algorithm can be implemented using non-pipelined and pipelined or subpipelined techniques. The non-pipelined implementations result in optimization of area at the cost of speed. Whereas pipelined or subpipelined implementations result in enhanced throughput with increased area utilization as pipelining enables processing of multiple blocks simultaneously. Abysmal analysis of the AES algorithm implies that the S-box substitution is the key component in creating confusion in the encryption process and hence attempts are being made to improve the quality of S-box. AES algorithm with S-box using GF(28) Galois Field inversions based on polynomial basis, using composite field arithmetic has been implemented by the authors, where the critical path delays are reduced using multiple stages of pipelining (Liu and Parhi, 2008). The authors presented S-box generation using combinational logic based truth table. In these, techniques such as Positive Polarity Reed-Muller structure or its variance, Sum of Product, Product of Sum, Twisted Binary Decision Diagram and Binary Decision Diagram are used. These techniques lead to high throughput but very poor area cost ratio. Various memory based S-box implementations use BRAMs, ROMs and LUTs to generate S-box. The memory based S-box generation results in reduced resource utilization but due to unbreakable delay reduces the throughput (Granado-Criado et al., 2010; Zhang and Wang, 2010). Chih Peng and Hwang (2008) proposed a CAM-based SubBytes scheme to achieve increased throughput AES architecture. The authors have presented design of S-box using second-order reversible Cellular Automata and the strength of S-box is evaluated using methods such as Bit

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

Independence Criterion (BIC), non-linearity, entropy and correlation immunity bias (Gangadari and Rafi Ahamed, 2016).

AES Algorithm: The AES algorithm performs operations on 128-bit plaintext and uses identical key for encryption as well as decryption. The AES algorithm processes facts obstruct of 128-bit parts and performs 10, 12 and 14 rounds of operations employing a cipher secret of duration 128-bits, 192-bits and 256-bits respectively. The algorithm operates on data block comprised of a 4 4 byte matrix known as the state. The essential procedures of AES algorithm are carried out on the state. The operations of AES Encryption algorithm with 128-bit key size are show in Fig. 1.

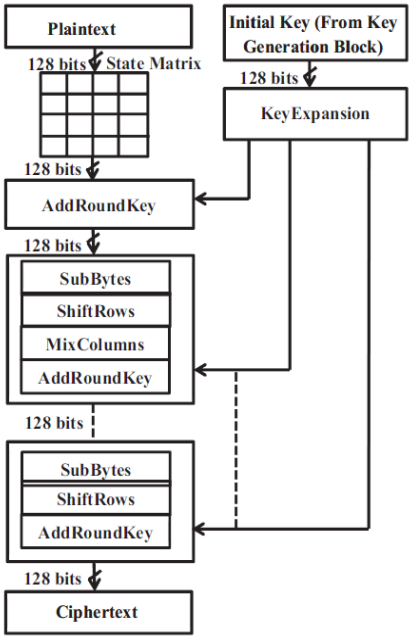

Figure 1: Flowchart of AES Algorithm.

The AES-128 algorithm can be divided in three stages viz. adding initial round key, rounds 1–9 and the final round. In the initial round, the 128-bit plaintext is Exclusive-ORed with 128-bit initial key. In each cipher round, SubBytes (), ShiftRows (), MixColumns () and AddRoundKeys () transformations are performed on a twodimensional 4 4 array of bytes called the states.

Design approach for AES algorithm with enhanced security: As compared to traditional AES algorithm, the proposed work suggests technique to modify the S-box values using PN Sequence Generator to improve the quality of encryption. The initial key required for encryption/decryption is also generated using the PN Sequence Generator instead of using a pre-defined key. In the proposed work, 8-bit PN Sequence Generator is used for generating the S-box values and initial key. The key and plaintext sensitivity tests are performed as per Strict Avalanche Criterion. The avalanche values for traditional AES algorithm are compared with avalanche values for AES algorithm with modified S-box values. For this comparison, pre-defined initial keys as well as initial keys generated using 8-bit PN Sequence Generator are considered. Further the proposed design is synthesized using different FPGA devices and comparison with existing FPGA implementations for speed and area optimization is done.

S-box values generation using 8-bit PN Sequence Generator A PN Sequence Generator is used to generate sequence of pseudorandom binary numbers. A PN Sequence Generator is designed using Linear Feedback Shift Register (LFSR) described by the Generator Polynomial. LFSR is shift register whose input bit is a linear function of previous state and

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

is generated by XORing selected bits from all the bits of the shift register. The number of states generated by the LFSR is determined by the feedback taps of the Generator Polynomial. The S-box of AES algorithm consists of 256 distinct 8-bit values. For generation of these S-box values, 8-bit PN Sequence Generator is used with maximal length feedback taps to generate 28 1 = 255 random values across (01)h to (FF) h. The value (00)h is randomly fed into the S-box. A random sequence with a very large repetition period is obtained by combining elements from taps of the shift register and giving a feedback to the input of the generator. The randomness in the output values generated from PN Sequence Generator depends not only on the feedback taps but also on the non-zero initial 8-bit seed value given to the generator. The change in the seed value shifts the starting value and changes the sequence of the generated values. This results in generating sequence known only to the designers. These values can then be used to form the S-box. The invertible S-box is responsible to strengthen the AES algorithm against various attacks. The lack of knowledge of the taps and seed value selected to the attackers will make the AES algorithm invulnerable to attacks. Key generation using 8-bit PN Sequence Generator In the proposed work, the key generation block uses 8-bit PN Sequence Generator to generate the initial key required for encryption/decryption process. The 8-bit PN Sequence Generator generates 255 values of 8-bit each which can be concatenated to form the 128-bit initial key. The key generation block internally selects the bytes for concatenation and form the 128-bit key. The output states (each of 8-bit) of the PN Sequence Generator are stored in the Look up Table (LUT) and are given as an input to 256:1 multiplexer. An 8-bit counter generates the value of 8-bit select lines of the multiplexer. The counter is designed to count 16 states so as to select 16 input bytes and form the 128-bit (16 8) value at the output. Thus the key will consists of 16 distinct bytes and a large number of distinct keys can be obtained by changing the initial value of the counter for select lines. These keys can then be dynamically applied to different blocks of 128-bit plaintext from the entire message to be encrypted. The generation of key using the key generation block eliminates the need of using external predefined keys. Moreover, the generated key is dependent on the feedback taps and initial seed value of the PN Sequence Generator and change in these parameters will change the value of initial key. In case an adversary tries to decipher the data using brute force attack, then due to use of a different key for each message, the attacker will be unable to decrypt any useful information from the message. Further, the brute force attack will identify the initial key, but without the knowledge of the S-box described in Section 4.1, the attacker will not be able to decipher the input text. Thus these two modifications ensure a very high encryption quality for the cipher.

III. IMPLEMENTATION

For hardware implementation of the proposed non pipelined AES Algorithm, Xilinx Spartan6 - FPGA device is used. The Xilinx ISE14.1 tool is used for Synthesis, simulation and generating the programmable file.

Performance evaluation parameters: The metrics for evaluating the performance of the proposed AES Algorithm with modified S-box and improved key generation technique are as mentioned below. Avalanche Effect: The strength of the cryptosystem is tested on the basis of Strict Avalanche Criterion (SAC). SAC is said to be satisfied whenever complementing a single input bit results in change of each of the output bits with a 50% probability. The plaintext and key each of 128-bit are the inputs to the AES algorithm. Thus, with a single bit complemented in plaintext or key, the cipher text should change with a probability of 50% known as Avalanche Effect.

Experimentation results: The proposed AES algorithm with modified S-box and improved key generation technique is evaluated on various FPGA devices for the performance parameters described in Section 5.1 using Xilinx ISE 14.1. The plaintext and key sensitivity on the Strict Avalanche Criterion is examined for 2048 variations using traditional AES algorithm and AES algorithm with modifications described in this work. From this, the average percentage avalanche value is calculated.

Figure 2 shows the comparison of Percentage Avalanche Effect for traditional and modified AES algorithm for the initial key and plaintext. As per the SAC, a single bit change in the input should result in 50% change in output bits. Thus, for calculating the avalanche effect all 128-bits from the initial key are complemented one by one and the corresponding ciphertext with traditional AES and the modified AES are obtained. The ciphertext generated for change in each input bit is compared with the ciphertext obtained with initial key for traditional AES. The count of number of

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

bits changed in the two values is divided by 128 and the value of percentage avalanche effect is calculated for the corresponding input. These steps are also followed for generating the value of percentage avalanche effect for remaining inputs of traditional AES and for modified AES. The plot for these values is shown in Figure 2. The overall percentage avalanche effect is calculated by taking the count of the number of values increased for modified AES. For the above mentioned set of key and plaintext, an increased percentage avalanche effect of 58% is achieved with modified AES algorithm. Similar experimentation is carried out on different sets of plaintext and initial keys generated using the technique discussed in Section. For 2048 variations in key and plaintext the percentage avalanche effect achieved varies in the range from 53% to 61% for modified AES algorithm. For performance evaluation, the proposed algorithm is synthesized on various FPGA devices using Xilinx ISE14.1. The proposed non-pipelined design is synthesized using different FPGA devices and its result are compared with existing nonpipelined implementations as shown in Table. Latency of 10 is considered while calculating the throughput of the proposed design as it generates the output after every 10 clock cycles. The proposed design achieves the highest throughput of 6.34Gbps on XC7VX690T device with maximum clock frequency of 495.32 MHz. The number of slices used is only 0.47% of the total available slices for the selected device.

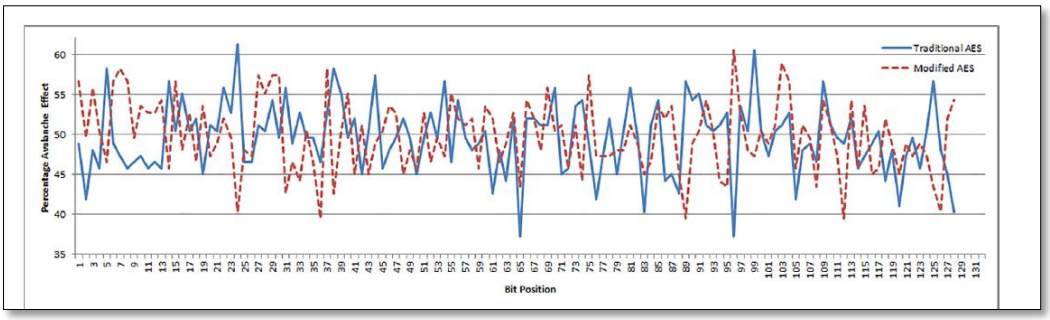

Figure 2: Comparison of % Avalanche Effect for traditional and modified AES algorithm

Compared to design which generates output after every 11 clock cycles, throughput of the proposed design is improved by 1.2 times at the expense of increased area usage by 1.67 times. Similarly an improvement in throughput by 1.8 times is achieved at the cost of 8.4 times more slice usage as compared to design. Similar is the case with (Wang and Ha, 2013). The proposed design improves the throughput by 1.8 and 1.33 times for the devices XC6VLX240T and XC4VLX60 respectively as compared to (Qiang et al., 2015). The implementation of proposed design on hardware using Xilinx Spartan6 XC6SLX150 device gives a throughput of 3.03 Gbps with maximum frequency of 237.45 MHz. The comparison of synthesis result of the proposed nonpipelined design with existing pipelined implementations is shown in Table.

Table: Comparison of proposed design with existing pipelined designs

Design	Device	No. of Pipeline stages	Bitwidth (bits)	No. of Slices	F_{max} (MHz)	Throughput (Gbps)	Mbps/Slice
Wang and Ha (2013)	XC6VLX240T	6	128	5927	319.29	40.87	6.90
Proposed	XC6VLX240T	Non-pipelined	128	4095	463.42	59.31	1.44
Henezen and Fichtner, (2010)	XC5VSX240T	2	128	1499	233.00	119.30	8.06
Proposed	XC5VSX240T	Non-pipelined	128	3420	199.18	25.50	7.45
Reddy and Praneeth, (2011)	XC5VLX110T	3	128	8896	202.26	25.89	2.91
Proposed	XC5VLX110T	Non-pipelined	128	3788	232.30	29.73	7.84

Compared to the 6-stage pipelined design, the proposed non-pipelined design improves the throughput by 1.45 times and utilizes 0.69 times less area. The throughput obtained by is more compared to proposed non pipelined design at the expense of pipelined design and four parallel Encryption datapath. However for future consideration, introducing pipelining in the proposed design will further improve the throughput of the proposed design. Reddy and Praneeth

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 8, September 2018*

International Conference on Emerging Trends in Science, Technology and Mathematics (ICETSTM-2018)"

22nd August 2018

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

(2011) have proposed design which uses complex combinational logic for SubBytes transformation and is 3-stage pipelined. In the proposed non-pipelined design LUT based SubBytes transformation is used which results in improving the throughput 1.15 times and utilizes 0.42 times less area as compared.

Conclusions: Encryption using hardware platforms is widely being used in order to secure data and enhance throughput. In this paper, techniques to enhance the encryption quality of AES algorithm and its implementation on FPGA are proposed. First, the S-box values in the modified AES algorithm are generated using PN Sequence Generator. Second, the initial key required for encryption/decryption is also based on the output of PN Sequence Generator. The result of encryption for modified AES algorithm is tested on the Strict Avalanche Criterion for 2048 variations and the average percentage avalanche effect of 60% is achieved for modified AES algorithm as compared to tradition AES algorithm. Thus the modifications suggested, results in improved quality of encryption. FPGAs are used for efficient hardware implementation of the modified AES algorithm. The results for throughput and area are compared with existing non-pipelined and pipelined designs and are found to achieve better performance. The proposed design achieves a throughput of 59.3Gbps when synthesized on XC6VLX240T device with a maximum frequency of 463.42 MHz and 30.39 Gbps when implemented on XC6SLX150 with a maximum frequency.

REFERENCES

1. Ali, L., Aris, I., Hossain, F.S., Roy, N., 2011. Design of an ultra high speed AES processor for next generation IT security. Comput. Electr. Eng. 37, 1160–1170.
2. Bogdanov, A., Khovratovich, D., Rechberger, C., 2011. Biclique cryptanalysis of the Full AES. Adv. Cryptol. ASIACRYPT 2011, 344–371.
3. Chodowiec, P., Gaj, K., 2003. Very Compact FPGA implementation of the AES algorithm. CHES 2003. LNCS 2779, 319–333.
4. Y. Wang, Y. Ha, "FPGA-Based 40.9-Gbits/s Masked AES with Area Optimization for Storage Area Network", Circuits and Systems II: Express Briefs, IEEE Transactions, vol.60, no.1, pp.36-40, Jan 2013.
5. D. Osvik, J. Bos, D. Stefan and D. Canright, "Fast Software AES Encryption", FSE'10 Proceedings of 17th International Conference on Fast Software Encryption, pp 75-93, 2010.
6. T. Babu, K. Murthy and G. Sunil, "AES Algorithm Implementation using ARM Processor", 2nd International Conference and workshop on Emerging Trends in Technology Proceedings published by IJCA, 2011
7. M. Hasamnis, P. Jambhulkar and S. Limaye, "Implementation of AES as a Custom", Advanced Computing: An International Journal (ACIJ), vol.3, No.4, July 2012. FIPS-197, "Federal Information Processing Standards Publication FIPS-197, Advanced Encryption Standard (AES)" http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, October 1999
8. N.Anitha Christy and P.Karthigaikumar, "FPGA Implementation of AES Algorithm using Composite Field Arithmetic", ICDCS, pp. 713-717, 2012.
9. FIPS-197, "Federal Information Processing Standards Publication FIPS-197, Advanced Encryption Standard (AES)" http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, October 1999
10. J. Bos, D. Osvik, D. Stefan, "Fast Implementations of AES on Various Platforms", IACR Cryptology ePrint Archive, vol. 501, 2009.
11. M. Biglari, E. Qasemi, B. Pourmohseni, "Maestro: A high performance AES encryption/decryption system", CADS, 17th CSI International Symposium, pp.145-148, 30-31 Oct. 2013.
12. Ebrahim, M., Khan, S., Khalid, U. Bin, 2014. Symmetric algorithm survey: a comparative analysis. Int. J. Comput. Appl. 61, 12–19.
13. Elbirt, A.J., Yip, W., Chetwynd, B., Paar, C., 2001. An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists. IEEE Trans. Very Large Scale Integr. Syst. 9, 545–557.
14. Farooq, U., Aslam, M.F., 2017. Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA. J. King Saud Univ. Comput. Inf. Sci. 29, 295–302. https://doi.org/10.1016/j. jksuci.2016.01.004. Fips-197, 2001.
15. Natl. Inst. Stand. Technol. 8– 12. https://doi.org/10.1016/S1353-4858(10)70006-4. Gaj, K., Chodowiec, P., 2009. FPGA and ASIC implementations of AES. Cryptogr. Eng. 235–294. https://doi.org/10.1007/978-0-387-71817-0.
16. Gangadari, B.R., Rafi Ahamed, S., 2016. Design of cryptographically secure AES like S-Box using second-order reversible cellular automata for wireless body area network applications. Healthcare Technol. Lett. 3, 177–183. https://doi.org/ 10.1049/htl.2016.0033.
17. Chih Peng, F., Hwang, J.K., 2008. FPGA implementations of high throughput sequential and fully pipelined AES algorithm. Int. J. Electr. Eng. 15, 447–455.
18. Chih-Pin, S., Tsung-Fu, L., Chih-Tsun, H., Cheng-Wen, W., 2003. A high-throughput low-cost AES processor. IEEE Commun. Mag., 86–91
19. Granado-Criado, J.M., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Gómez-Pulido, J.A., 2010. A new methodology to implement the AES algorithm using partial and dynamic reconfiguration. Integr. VLSI J. 43, 72–80.
20. Hammad, I., El-Sankary, K., El-Masry, E., 2010. High-speed AES encryptor with efficient merging techniques. IEEE Embedded Syst. Lett. 2, 67–71. https://doi. org/10.1109/LES.2010.2052401.
21. Henzen, L., Fichtner, W., 2010. FPGA parallel-pipelined AES-GCM core for 100G ethernet applications. Proc. ESSCIRC, 202–205.
22. Hongge, L., Jinpeng, D., Yongjun, P., 2012. Cell array reconfigurable architecture for high-efficiency AES system. Microelectron. Reliab. 52, 2829–2836. https://doi. org/10.1016/j.microrel.2012.04.020.
23. Hussain, U., Jamal, H., 2012. An efficient high throughput FPGA implementation of AES for multi-gigabit protocols. Proc. 10th Int Conf. Front. Inf. Technol. FIT 2012, 215–218.
24. Liu, L., Luke, D., 2003. Implementation of AES as a CMOS core. CCECE 2003. Can. Conf. Electr. Comput. Eng. 1, 53–56.
25. Liu, R., Parhi, K.K., 2008. Fast composite field S-box architectures for advanced encryption standard. ACM Gt. Lakes Symp. VLSI, 65–70