ISSN: 0974-5823 Vol. 7 No. 4 April, 2022

International Journal of Mechanical Engineering

VLSI Implementation of Modified AES System for FPGA-IOT Application

Chandrajeet Singh[1], Prof. Ashish Raghuwanshi[2]

[1]M.Tech Scholar, [2]Assistant Professor,

Department of Electronics and Communication Engineering

IES College of Technology, Bhopal, India

Abstract

Advance Encryption Standard (AES) is considered as one of the secure and efficient algorithms. Despite that like other symmetric encryption algorithms, the secret key distribution is still considered as a critical issue. Again to encrypt or decrypt a single block (128-bit) of data, an essential amount of computational processing has to be done which consumes more power. Internet of things (IoT) is the extension of the Internet to connect just about everything on the planet. This paper presents implementation of data security algorithm based on modified advanced encryption standard with 256 bit key for IOT application. MAES is a lightweight version of AES which meets the demand. A new one-dimensional substitution Box (S-box) is proposed instead of conventional 2-D S-box and previous 1-D S-box. Simulated result shows that proposed MAES gives better performance than previous MAES in term of delay, throughput, transmission time, efficiency rate.

Keywords: *IOT, Wireless, Security, Cryptography, Encryption, Decryption, Block Cipher, Simulation, Synthesis, Xilinx.*

1. Introduction

5G is the fifth era of cell portable interchanges. It succeeds the 4G (LTE/WiMax), 3G (UMTS) and 2G(GSM) frameworks. Internet of Things security is the area worried about ensuring interconnected gadgets and systems in the biological community. In an IoT biological system registering gadgets and installed frameworks, likewise called things can impart information over system as they are furnished with special identifiers and capacity to gather, send and get information. IoT applications can be found in all divisions extending from home apparatuses to mechanical machine-to-machine (M2M) to shrewd vitality matrices. The individual data gathered and put away with these gadgets, for example, your name, age, wellbeing information, area and that's just the beginning can help crooks in taking your personality. In the meantime, the internet of Things is a developing pattern, with a surge of new items hitting the market. Be that as it may, here's the issue: When you're associated with everything, there are more approaches to get to your data. That can make you an alluring focus for individuals who need to make a benefit off of your own data. Every associated gadget you possess can include another protection concern, particularly since the vast majority of them interface with your cell phone. Here's the manner by which it works. Regardless of whether you have to check the cameras in your home, bolt or open an entryway, modify temperature or lighting, pre-warm the broiler, or kill a television, you can do everything remotely with only a couple of taps on your cell phone. Be that as it may, the more functionality you add to your cell phone, the more data you store in the gadget. This could make cell phones and anything associated with them helpless against a huge number of various kinds of assaults. The exchange of digital data over a network has exposed the multimedia data to various kinds of abuse such as Brute-Force attacks, unauthorized access, and network hacking. Therefore, the system must be safeguarded with an efficient media-aware security framework such as encryption methods that make use of standard symmetric encryption algorithms, which will be responsible for ensuring the security of the multimedia data. For the encryption of electronic data, one of the most prominent cryptographic algorithms is the Advanced Encryption Standard algorithm.

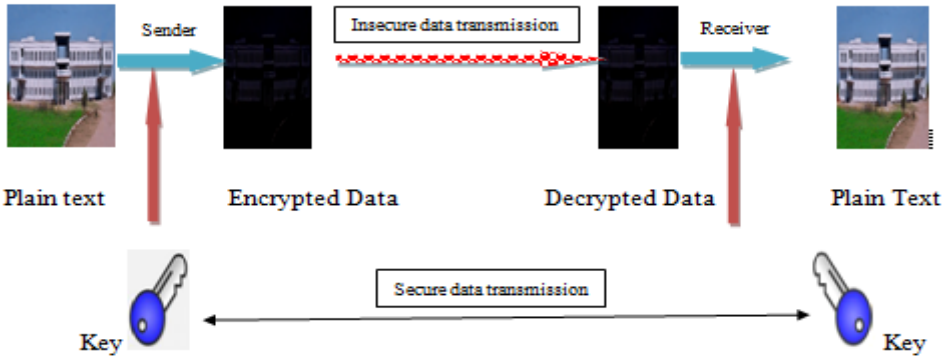

Figure 1: Basic block diagram of cryptography process

The Advanced Encryption Standard (AES) has been lately accepted as the symmetric cryptography standard for confidential data transmission. However, the natural and malicious injected faults reduce its reliability and may cause confidential information leakage. In this paper, we study concurrent fault detection schemes for reaching a reliable AES architecture. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. Generally speaking, it uses a cryptographic system to transform a plaintext into a cipher text, using most of the time a key. As networking technology advances, the gap between network bandwidth and network processing power widens. Information security issues add to the need for developing high-performance network processing hardware, particularly that for real-time processing of cryptographic algorithms.

AES is basically a security algorithm is used for encryption and decryption of data. Encryption is the process in which we perform a fixed set of operation on the data to randomize the data and transform it into some meaningless form so that even if any unauthorized agents gets an access of the data, will not be able to obtain the useful information present in the data. Such data which is apparently meaningless is transmitted. Such data can be converted back to its useful form, that is, the actual data only with the key of the key at the receiving end. Keys are basically a secret binary data of above said fixed length which are used to encrypt (cipher) the original data at the transmitting end to obtain the encrypted data and decrypt (de-cipher) the encrypted data to get back the original data at the receiving end. Its obvious that the key with the help of which the data will be retrieved at the receiving end will be known at the receiving end prior to the establishment of the communication. This process of retrieving the original data is called decryption. The branch of science which deals with encryption and decryption of data is known as Cryptography. The algorithms with the help of which we implement encryption or decryption of data are called Cryptographic algorithms.

2. Methodology

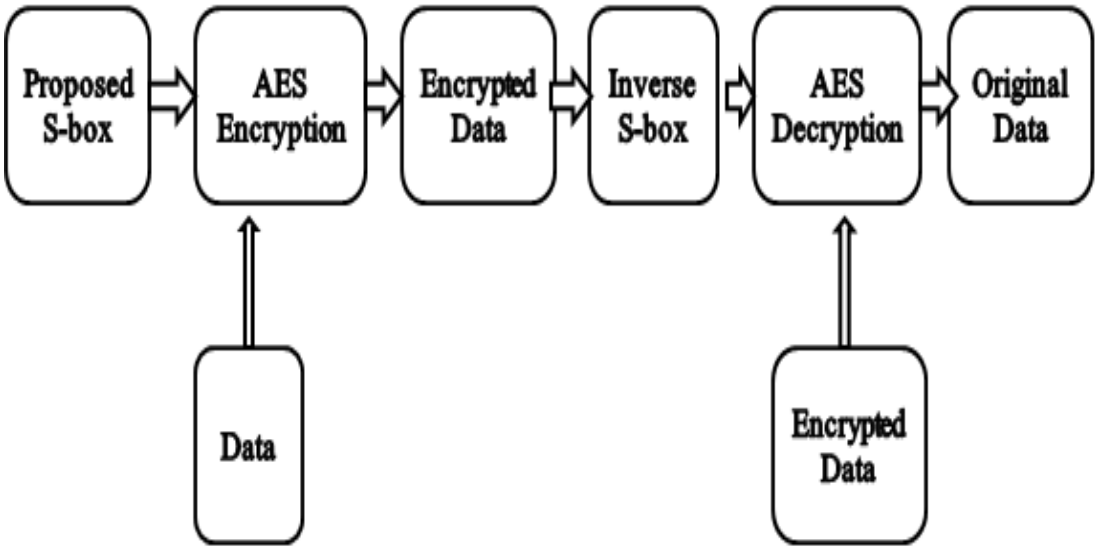

Figure 2: Flow Chart

Cryptographic algorithms can be either symmetric or non-symmetric. Symmetric Cryptographic algorithms are those in which we use the same set of keys both at the transmitting end as well as the receiving end. AES is a symmetric block cipher. AES Algorithm may be used with the three different key lengths of 128,192 and 256. AES is referred to as "AES-128", "AES-192", and "AES-256" accordingly. In the proposed work we have used AES-128. Thus, symmetric cipher requires a single key for both encryption and decryption, which is independent of the plaintext and the cipher itself. Hence, it would be impractical to retrieve the plaintext solely based on the cipher text and the decryption algorithm, without knowing the encryption key. Thus, the secrecy of the encryption key is of high importance in symmetric ciphers such as AES.

AES can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, or 256 bits. In the proposed work, the key length is 128 bits. Rijndael was designed to handle additional block sizes and key lengths, and however they are not adopted in this standard. The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext. The 128 bit data block is divided into 16 bytes. These bytes are mapped to a 4×4 array called the state and all the internal operation can be performed on state. Internally, the AES algorithm's operations are performed on a two-dimensional array of bytes called the State. The encryption process includes the following transformations of states: SubBytes(), ShiftRows(), MixColumns(), and AddRoundKey(). The encryption process also includes a key schedule. The AES algorithm takes the Cipher Key, K, and performs a Key Expansion routine to generate a key schedule. In the decryption process, the Cipher transformations are inverted and then implemented in reverse order to produce a straightforward Inverse Cipher for the AES algorithm. The individual

transformations used in the Inverse Cipher are InvShiftRows(), InvSubBytes(), InvMixColumns(), and AddRoundKey() . The decryption process also includes a key schedule similar to Encryption process.

In this paper, we have implemented WiMax/IOT Security using Modified Advanced Encryption Standard Cryptographic Algorithm. It is designed WiMax MAES Security Algorithm sub-module, both at the Encryption and Decryption end, based on the internal operations of the algorithm, as mentioned above. Each sub-module is designed, simulates and synthesized step by step as per algorithm. The results of simulation and synthesis are presented separately.

3. Simulation Results

The designed WiMax/IOT MAES Security Algorithm implementation has multiple sub-modules inside it both at the Encryption and Decryption end, based on the internal operations of the algorithm. Top module is designed, simulated and synthesized as per proposed algorithm. First we are presenting the results of simulation.

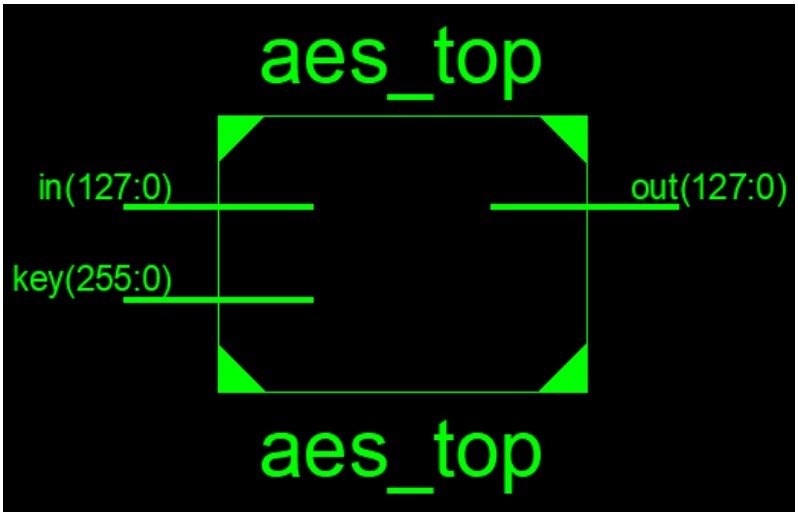

Figure 3: Top View of proposed Modified AES

In figure 3, top view of proposed Modified AES algorithm, where 128 bit input, 128 bit output and 256 Encryption and 256 Decryption key taken.

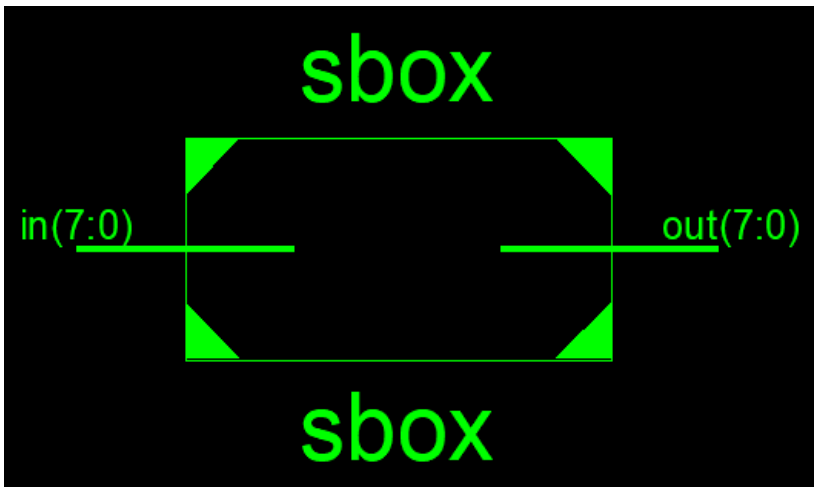

Figure 4: Top module of proposed 1D S-box

Figure 4 is showing the top module of the proposed 1 dimensional sub-byte box. Here 8bit input is giving to the Sbox and its generating 8 bit output after operation of s-box.

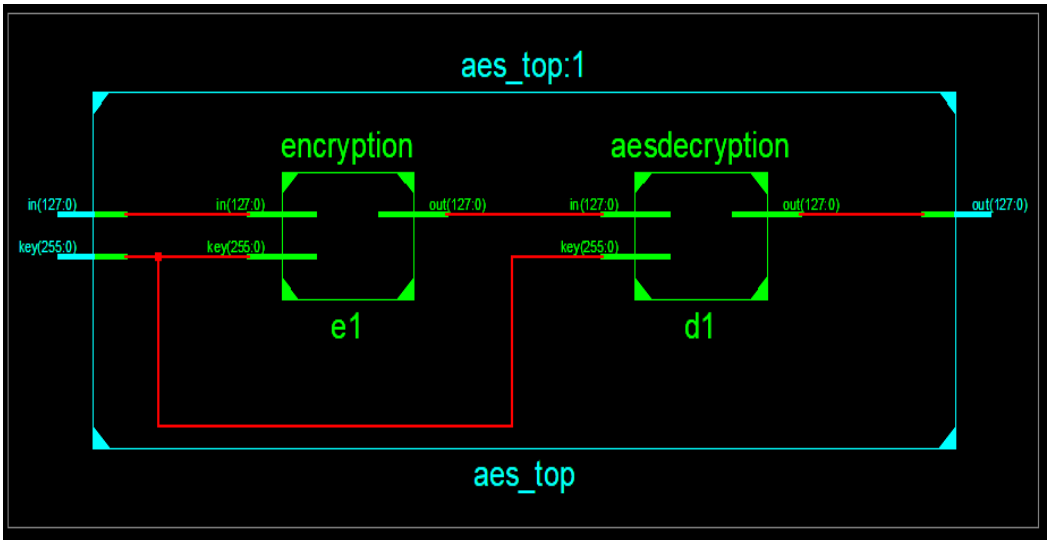

Figure 5: RTL view of Encryption and Decryption Process

The figure 5 is showing the RTL view of encryption and decryption process. The 128 bit input data is encrypted by the 256 bit key and at the output side it is decrypted by same 256 bit key and original data is recovered.

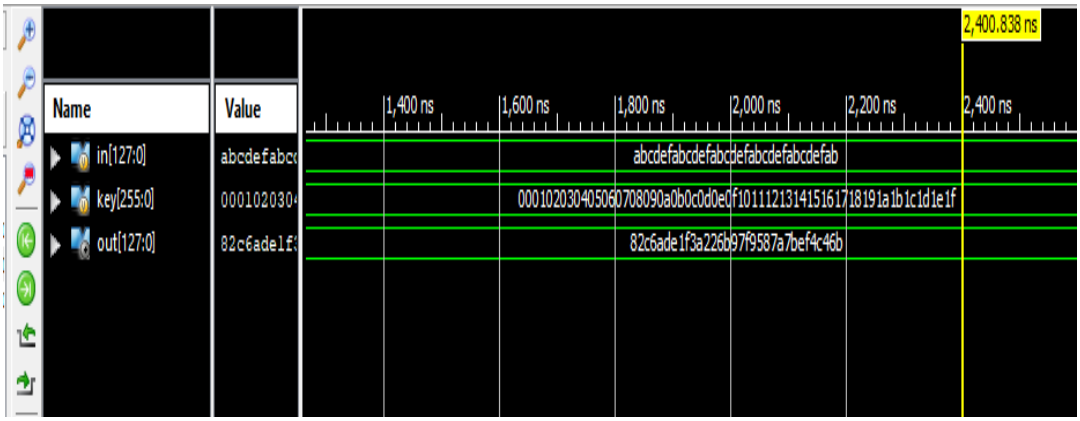

Figure 6: MAES Encryption process

Figure 6 presents the encryption process of the proposed modified AES algorithm.

Input – abcdefabcdefabcdefabcdefabcdefab

Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

Output - 82c6ade1f3a226b97f9587a7bef4c46b

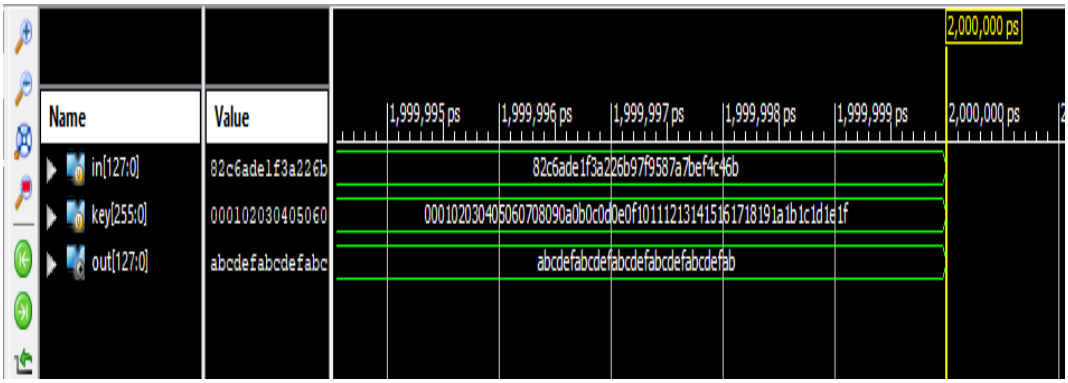

Figure 7: MAES Decryption Process

Figure 7 presents the decryption process of the proposed modified AES algorithm.

Input – 82c6ade1f3a226b97f9587a7bef4c46b

Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

Output - abcdefabcdefabcdefabcdefabcdefab

Table 1: Comparison of Simulation Results

Sr No.	Parameters	Previous Result [1]	Proposed Result
1	S- box components	86	16
2	Combinational AES components	19,120	13016
3	S box delay (ns)	6.830	3.12
4	Combinational AES delay (ns)	20	13.49

4. Conclusion

This paper presents implementation of data security algorithm based on modified advanced encryption standard with 256 bit key for IOT application. It is developed for the implementation of both encryption and decryption process. The S- box components of proposed work is 16 while previous it is 86. The combinational AES components of previous is 19,120 while proposed is 13016. The S box delay by previous is 6.830 ns while proposed it is 3.12 ns. The combinational AES delay is 20ns by previous and while proposed it is 13.49ns. Therefore the simulation results achieved significant better performance than the existing research work.

References

1. *T. B. Singha, R. P. Palathinkal and S. R. Ahamed, "Implementation of AES Using Composite Field Arithmetic for IoT Applications," 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP), 2020, pp. 115-121, doi: 10.1109/ISEA-ISAP49340.2020.235009.*
2. *S. S. S. Priya, P. Karthigai Kumar, N. M. SivaMangai and V. Rejula, "FPGA implementation of efficient AES encryption,"* 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), *Coimbatore, 2015, pp. 1-4.*
3. *R. V. Kshirsagar and M. V. Vyawahare, "FPGA Implementation of High Speed VLSI Architectures for AES Algorithm,"* 2012 Fifth International Conference on Emerging Trends in Engineering and Technology, *Himeji, 2012, pp. 239-242.*
4. *Abhiram L S, Sriroop B K, Gowrav L, Punith.Kumar H L and M. C. Lakkannavar, "FPGA implementation of dual key based AES encryption with key Based S-Box generation,"* 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), *New Delhi, 2015, pp. 577-581.*
5. *M. El Maraghy, S. Hesham and M. A. Abd El Ghany, "Real-time efficient FPGA implementation of aes algorithm,"* 2013 IEEE International SOC Conference, *Erlangen, 2013, pp. 203-208.*
6. *T. Phan, V. Hoang and V. Dao, "An efficient FPGA implementation of AES-CCM authenticated encryption IP core,"* 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), *Danang, 2016, pp. 202-205.*
7. *S. S. H. Shah and G. Raja, "FPGA implementation of chaotic based AES image encryption algorithm,"* 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), *Kuala Lumpur, 2015, pp. 574-577.*
8. *S. Qu, G. Shou, Y. Hu, Z. Guo and Z. Qian, "High Throughput, Pipelined Implementation of AES on FPGA,"* 2009 International Symposium on Information Engineering and Electronic Commerce, *Ternopil, 2009, pp. 542-545.*
9. *P. N. Khose and V. G. Raut, "Implementation of AES algorithm on FPGA for low area consumption,"* 2015 International Conference on Pervasive Computing (ICPC), *Pune, 2015, pp. 1-4.*
10. *N. Gaur, A. Mehra and P. Kumar, "Enhanced AES Architecture using Extended Set ALU at 28nm FPGA,"* 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), *Noida, 2018, pp. 437-440.*
11. *J. Senthil Kumar and C. Mahalakshmi, "Implementation of pipelined hardware architecture for AES algorithm using FPGA,"* 2014 International Conference on Communication and Network Technologies, *Sivakasi, 2014, pp. 260-264.*
12. *A. M. Atteya and A. H. Madian, "A hybrid Chaos-AES encryption algorithm and its impelmention based on FPGA,"* 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS), *Trois-Rivieres, QC, 2014, pp. 217-220.*
13. *R. Paul and S. Shukla, "Partitioned security processor architecture on FPGA platform," in* IET Computers & Digital Techniques, *vol. 12, no. 5, pp. 216-226, 9 2018.*
14. *R. Lumbiarres-López, M. López-García and E. Cantó-Navarro, "Hardware Architecture Implemented on FPGA for Protecting Cryptographic Keys against Side-Channel Attacks," in* IEEE Transactions on Dependable and Secure Computing, *vol. 15, no. 5, pp. 898-905, 1 Sept.-Oct. 2018.*
15. *Q. Liu, Z. Xu and Y. Yuan, "High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion," in* IET Computers & Digital Techniques, *vol. 9, no. 3, pp. 175-184, 5 2015.*