Design and Analysis of Modified S-Box Based AES Security Algorithm for IOT Application

Lalan Kumar Jha[1], Dr. Anshuj Jain[2]
[1]M.Tech Scholar, [2] Associate Professor & Head,
[1,2]Department of Electronics & Communication Engineering,
[1,2] SCOPE College of Engineering, Bhopal (M.P.), India.

Abstract : Internet of things (IoT) is the extension of the Internet to connect just about everything on this earth. Security of multimedia data is an imperative issue because of fast evolution of digital data exchanges over unsecured network. Multimedia data security is achieved by methods of cryptography, which deals with encryption of data. Most of the application uses Advanced Encryption Standard (AES) and modify it, to reduce the calculation of algorithm and for improving the encryption performance. In modified AES algorithm used 1-Dimensionla S-box instead of 2-Dimensional S-box. Theoretical analysis and experimental results prove that this technique provides high speed as well as low latency on data encryption and decryption. Modified-AES algorithm is a fast lightweight encryption algorithm for security of multimedia data. Achieved efficiency of proposed MAES is around 23.7% while previously it is 18.35%. The total latency of previous work is 29.983milli sec while this work achieved 38.161ns.

IndexTerms **–**IOT, Block Cipher, Security, Cryptography, Encryption, Decryption, , Simulation, Xilinx.

I. INTRODUCTION

The Advanced Encryption Standard (AES) has been recently acknowledged as the symmetric cryptography standard for private information transmission. Nonetheless, the normal and malevolent infused shortcomings diminish its unwavering quality and may cause classified data spillage. In this paper, we study simultaneous flaw identification plans for arriving at a solid AES engineering. Cryptography is the study of mystery codes, empowering the secrecy of correspondence through a shaky channel. It ensures against unapproved parties by avoiding unapproved adjustment of utilization. As a rule, it utilizes a cryptographic framework to change a plaintext into a cipher text, utilizing more often than not a key. As systems administration innovation progresses, the hole between system transfer speed and system handling force enlarges. Data security issues add to the requirement for growing elite system handling equipment, especially that for constant preparing of cryptographic calculations.

AES is fundamentally a security calculation is utilized for encryption and unscrambling of information. Encryption is the procedure where we play out a fixed arrangement of activity on the information to randomize the information and change it into some inane structure so that regardless of whether any unapproved operators gets an entrance of the information, won't almost certainly acquire the valuable data present in the information. Such information which is obviously inane is transmitted. Such information can be changed over back to its valuable structure, that is, the real information just with the key of the key at the less than desirable end. Keys are fundamentally a mystery parallel information of above said fixed length which are utilized to encode (cipher) the first information at the transmitting end to get the scrambled information and decode (de-cipher) the scrambled information to get back the first information at the less than desirable end. Clearly the key with the assistance of which the information will be recovered at the less than desirable end will be known at the less than desirable end before the foundation of the correspondence. This procedure of recovering the first information is called unscrambling. The part of science which manages encryption and unscrambling of information is known as Cryptography. The calculations with the assistance of which we actualize encryption or unscrambling of information are called Cryptographic calculations.

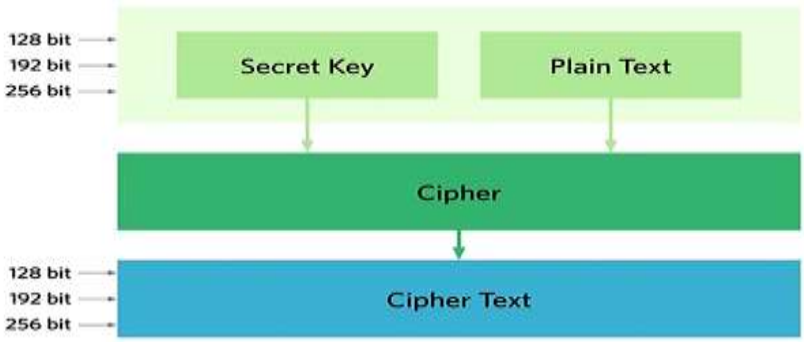

Figure 1: AES Model

II. PROPOSED APPROACH

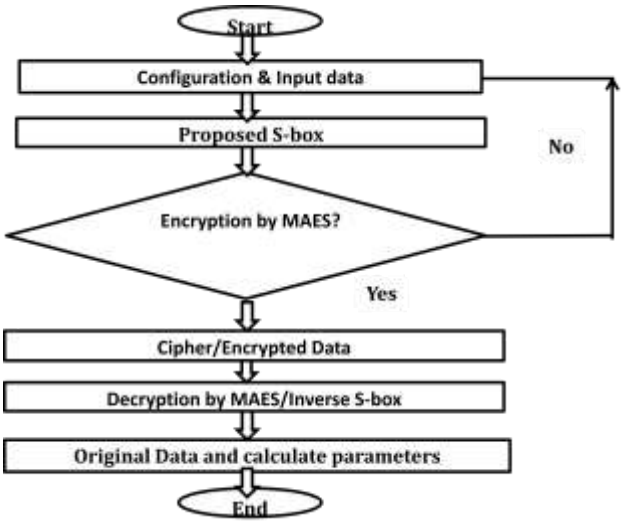

Figure 2: Flow Chart

Designed Proposed MAES using 128 bit input key and 256 bit secure key. 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.

III. SIMULATION RESULT

The designed 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. Now presenting the results of simulation.

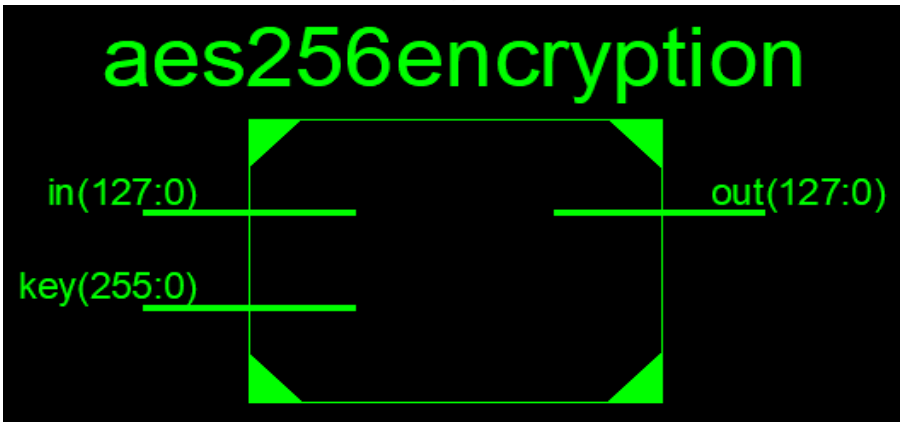

Figure 3: RTL Schematics of WiMax/IOT MAES

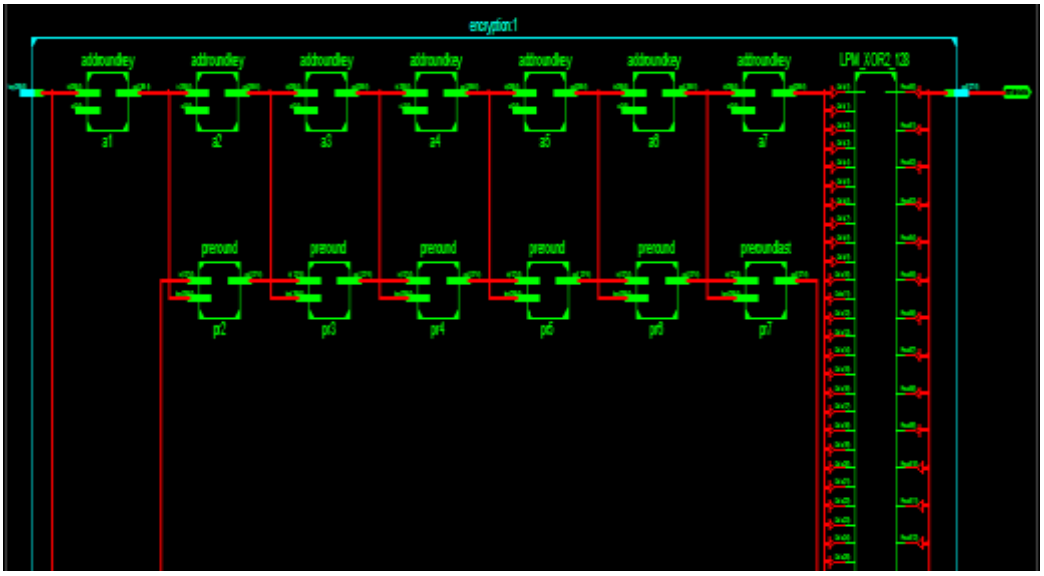

Figure 4: RTL view of proposed MAES algorithm

It is designed WiMax/IOT Security using Advanced Encryption Standard Cryptographic Algorithm. We have used Verilog for this purpose. We have used Xilinx ISE which has given synthesis results, as summarized in Table 1. Below. Also, we have depicted the pictorial representation of the results, which is the screenshot of tool generated Design Summary of individual sub-modules both at the WiMax data Encryption end and WiMax data Decryption end.

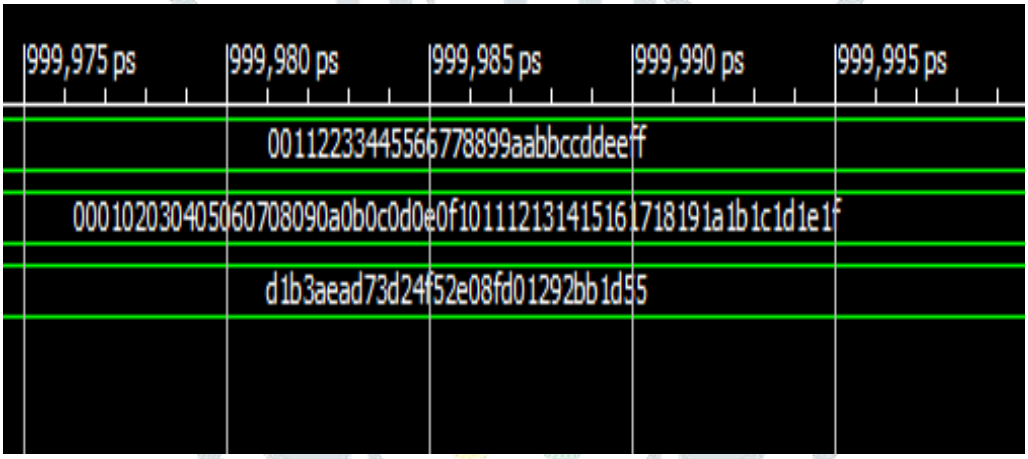

Figure 5: Encryption Process

In figure 5, firstly take 128 bit in input, in hexa form it is 00112233-445566778899aabbccddeeff. Then encrypted with secure key and generate cipher form of data.

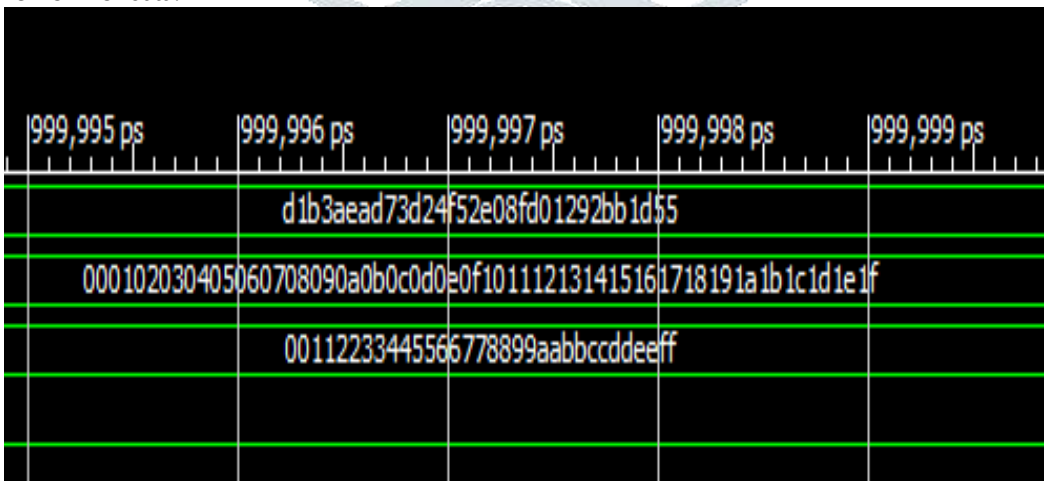

Figure 6: Decryption Process

In figure 6, take 128 bit in input, in hexa form it is d1b3aead73d24f524f2e08fd1292bb1d55. Then decrypted with inverse secure key and generate plain input i.e. 00112233-445566778899aabbccddeeff.

Table 1: Result Comparison of Proposed work with previous work

Sr No.	Parameters	Previous Result	Proposed Result
1	Software	nesC	Xilinx 14.7
2	Language	Structured programming	Verilog
3	Transmission time	4969.896 milli sec	2903.02 milli sec
4	Latency	29.983 milli sec	38.161ns
5	Packet	1000	1000
6	Efficiency rate	18.35%	23.7%
7	Voltage	1.5 V	1.2V

These comparison tables present various parameters values of previous work and proposed work. Existing work used nesC software while proposed work implemented using xilinx14.7 software. Verilog programming is used instead of structural programming. Latency, delay and transmission time is reduced than previous work. Therefore it is clear from comparison table that proposed work parameters gives significant good value than existing achieved values.

Xilinx contain various family or IC generation like Spartan, vertex kintex etc. Proposed 256 bit MAES check in various family in terms of delay, total memory, area, power and global maximum fanout. So it is clear that vertex 7 is advanced FPGA IC so it gives better outcome than other family.

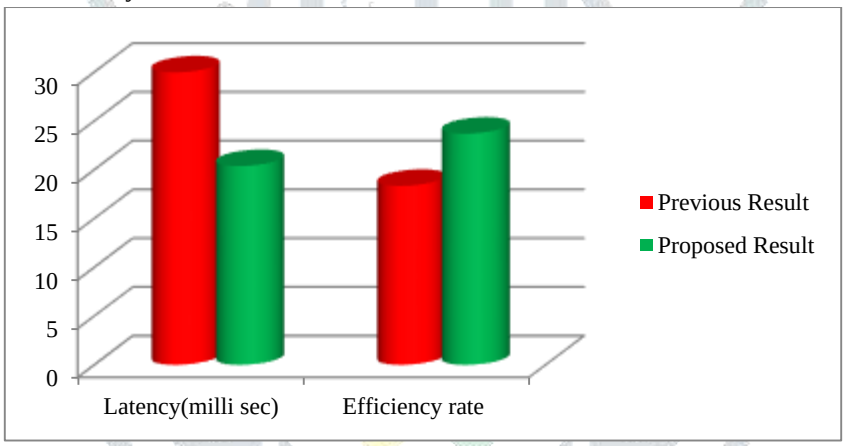

Figure 7: Latency and Efficiency comparison

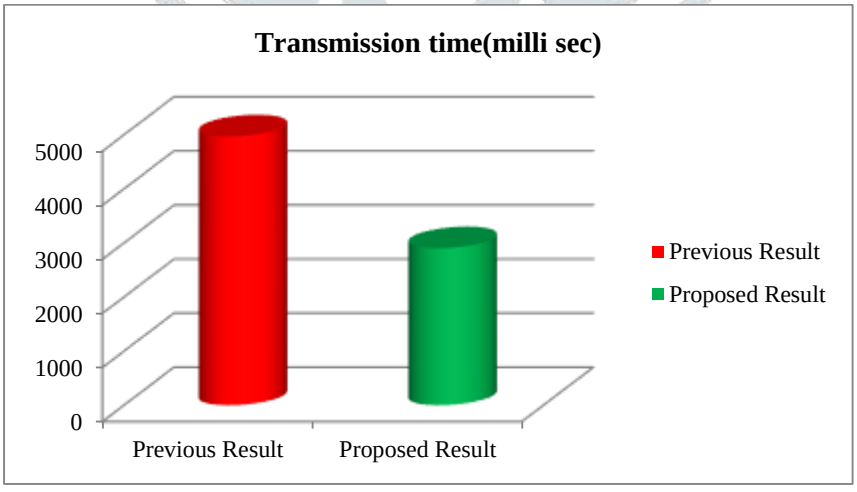

Figure 8: Transmission time comparison

Figure 7 & 8 showing graphical representation of latency efficiency and transmission time comparison. Therefore it is clear that proposed MAES gives better performance than previous MAES.

IV. CONCLUSION

This paper presents Modified advance encryption standard for 256 bit key with 1-D S-box. It is optimized and Synthesizable using verilog code in Xilinx software. It is developed for the implementation of both encryption and decryption process. Each

program is tested with some of the random values and output results are perfect with minimal delay. Therefore, MAES can indeed be implemented with reasonable efficiency on an FPGA, with the encryption and decryption taking an average of 32 and 34 ns respectively (for every 128 bits). MAES, a lightweight version of Advanced Encryption Standard (AES) which meets the demand. A new one-dimensional Substitution Box is proposed by formulating a novel equation for constructing a square matrix in affine transformation phase of MAES. Efficiency rate of MAES is around 23.7% in terms of packet transmission which indicates MAES consumes less energy than AES, 38.117ns latency achieved and it can be applicable for Internet of Things application.

REFERENCES

[1]. A. R. Chowdhury, J. Mahmud, A. R. M. Kamal and M. A. Hamid, "MAES: Modified advanced encryption standard for resource constraint environments," *2018 IEEE Sensors Applications Symposium (SAS)*, Seoul, 2018, pp. 1-6.

[2]. 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.

[3]. 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.

[4]. 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.

[5]. 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.

[6]. 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.

[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]. 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.

[9]. 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.

[10]. 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.

[11]. Bilgin, B. Gierlichs, S. Nikova, V. Nikov and V. Rijmen, "Trade-Offs for Threshold Implementations Illustrated on AES," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1188-1200, July 2015.

[12]. 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.

[13]. M. S. Kumar and S. Rajalakshmi, "Notice of Violation of IEEE Publication Principles
High efficient modified mixcolumns advanced encryption standard using Vedic multiplier," *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014*, Coimbatore, 2014, pp. 462-466.

[14]. 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.

[15]. A. Abed and A. A. Jawad, "FPGA implementation of a modified advanced encryption standard algorithm," *2013 International Conference on Electrical Communication, Computer, Power, and Control Engineering (ICECCPCE)*, Mosul, 2013, pp. 46-51.