International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 08, August 2022)

A VLSI Implementation of Modified S-Box with Reduced Area and Latency

Prayasun Bisandre[1], Prof. Shraddha Shrivastava[2]

[1]*Research Scholar*, [2]*Assistant Professor, Department of Electronics and Communication Engineering, Lakshmi Narain College of Technology, Bhopal, India*

***Abstract*— Cryptography plays an important role in the security of data transmission. 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. This paper presents VLSI implementation of modified S-Box with reduced area and latency. 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 the proposed modified S-box gives better performance than previous S-box in term of delay, throughput, and area.**

***Keywords*— Cryptography, Encryption, Decryption, S-box, Modified, Cipher, Simulation, Synthesis, Xilinx.**

I. INTRODUCTION

To protect the data transmission over insecure channels two types of cryptographic systems are used: Symmetric and Asymmetric cryptosystems. Symmetric cryptosystems such as Data Encryption Standard (DES), DES, and Advanced Encryption Standard (AES), uses an identical key for the sender and receiver; both to encrypt the message text and decrypt the cipher text. Asymmetric cryptosystems such as Rivest-Shamir- Adleman (RSA) & Elliptic Curve Cryptosystem (ECC) uses different keys for encryption and decryption. Symmetric cryptosystem is more suitable to encrypt large amount of data with high speed.

In general, an S-box takes some number of input bits, m, and transforms them into some number of output bits, n, where n is not necessarily equal to m.[1] An m×n S-box can be implemented as a lookup table with 2m words of n bits each. Fixed tables are normally used, as in the Advance Encryption Standard (AES), but in some ciphers the tables are generated dynamically from the key (e.g. the Blowfish and the Twofish encryption algorithms).

XSL ciphers are restricted class of Substitution-Affine ciphers. Their rounds are composed of the XOR of key material, a nonlinear substitution provided by an S-box, and a linear diffusion stage.

Before start looking into the algebraic properties of S-boxes in Rijndael and Serpent, which are exploited by XSL attack.

Rijndael encryption routine consists of 10..14 rounds. All rounds in succession are similar. The plaintext is fed through an XOR function, against a round key, of 128 - 256 bits. The bits of key material are fed into the side of XOR routines. These bytes are then utilized as a mapping index, for identical S-boxes, which maps inputs of 8 bits to outputs of 8 bits. The arrangement of bytes is then altered in a particular order. These bytes are then mixed via a linear function grouped in four.

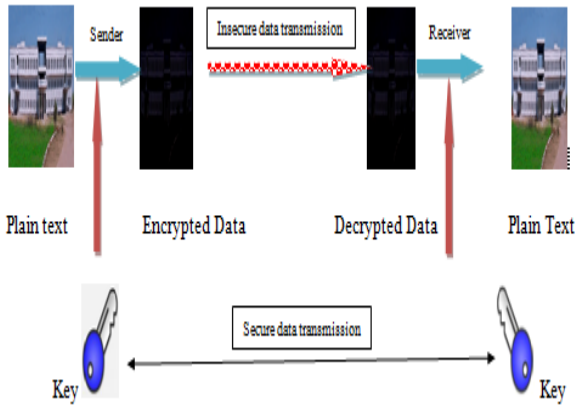

Figure 1: Basic block diagram of cryptography process

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.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 08, August 2022)

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.

II. METHODOLOGY

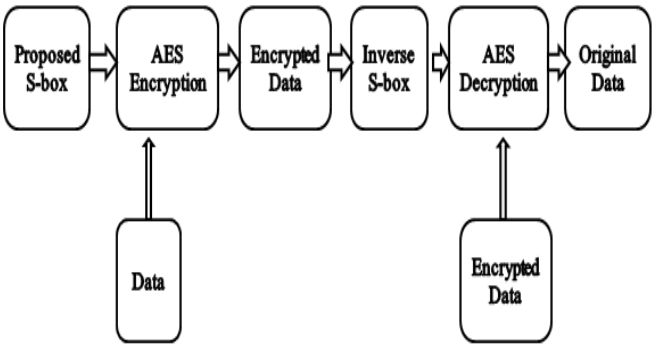

Figure 2: Flow Chart

Design One-dimensional S-box for proposed algorithm instead of two dimensional S-box in conventional advance encryption standard algorithm.

The procedure to generate the dynamic S-box is explained in de- tail in Chapter3. Choose an arbitrary key of size 16 bytes, which must be known at the receiver end also to decrypt the message. Construct a 4 × 4 matrix with these 16 bytes. Test for the sin- gularity of the matrix with respect to an irreducible polynomial which is also key dependent as in dynamic S-box construction. If the matrix is singular then XOR each byte of the key with [01h], then test for the singularity. Continue the same procedure till it is get a non-singular matrix.

Then use this matrix to perform the mix column operation as in the original AES encryption algorithm.

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.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 08, August 2022)

III. Simulation Results

The designed WiMax/IOT modified S-box 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.

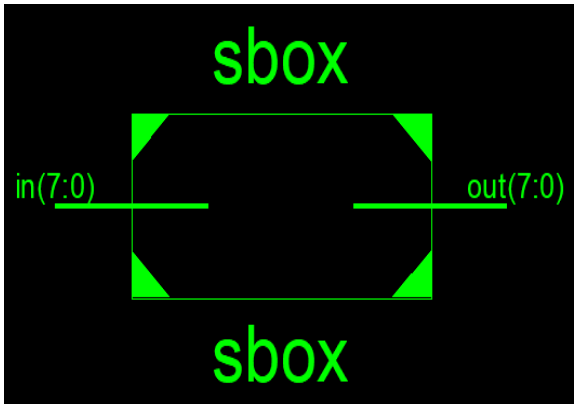

Figure 3: Top module of proposed 1D S-box

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. It 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.

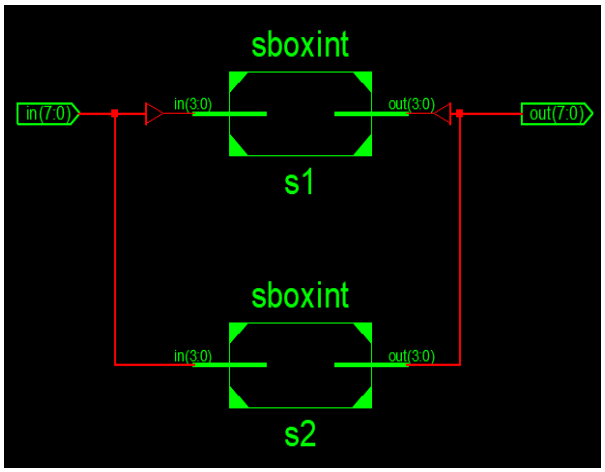

Figure 4: Internal view of proposed S-box

The figure 4 is showing the internal view of RTL of s-box parallel processing. 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.

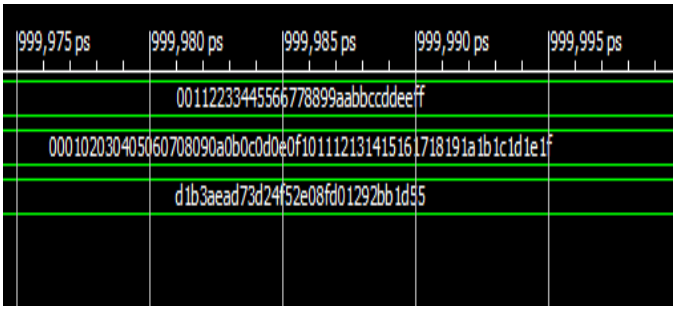

Figure 5: Encryption process

Figure 5 presents the encryption process of the proposed algorithm.

Firstly take 128 bit in input, in hexa form it is 00112233-445566778899aabbccddeeff.

Then encrypted with secure key and generate cipher form of data.

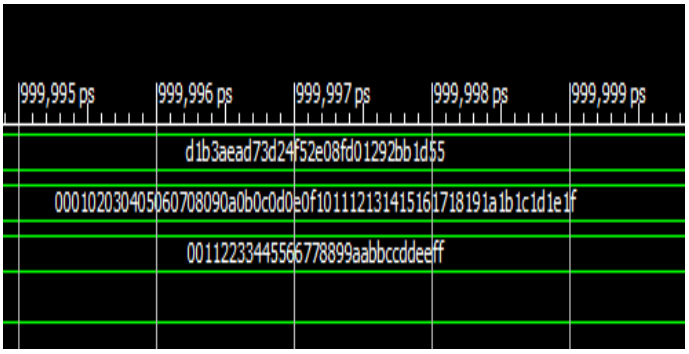

Figure 6: Decryption Process

Figure 6 presents the decryption process of the proposed algorithm.

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.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	8	204000	0%
Number of fully used LUT-FF pairs	0	8	0%
Number of bonded IOBs	16	600	2%

Figure 8: Device Utilization Summary of S-box

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 08, August 2022)

Table 1:
Result Comparison

Sr No.	Parameters	Previous Result [1]	Proposed Result
1	No of slices	31	8
2	Frequency	724.638 Mhz	1204 Mhz
3	Delay	1.379 ns	0.83ns
4	Throughput	5.79 Gbps	9.6 Gbps

IV. CONCLUSION

This paper presents implementation of data security algorithm based on modified s-box advanced encryption standard with 256 bit key for IOT application. It is developed for the implementation of both encryption and decryption process. The number of slices using by the previous work is 31 while in the proposed work; it is 8. The frequency of the proposed work is 1204 Mhz while in the previous it is 724.638 Mhz. The overall optimized delay is 0.83ns by the proposed and 1.379 ns by the previous. The data speed by the proposed work is 9.6 Gbps by the proposed work while 5.79 Gbps by the previous work. It is clear from the comparison table the proposed work gives better simulation results then previous.

REFERENCES

[1] Y. T. Teng, W. -L. Chin, D. -K. Chang, P. -Y. Chen and P. -W. Chen, "VLSI Architecture of S-Box With High Area Efficiency Based on Composite Field Arithmetic," in IEEE Access, vol. 10, pp. 2721-2728, 2022, doi: 10.1109/ACCESS.2021.3139040.

[2] N. Su, Y. Zhang and M. Li, "Research on Data Encryption Standard Based on AES Algorithm in Internet of Things Environment," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2018, pp. 2071-2075

[3] 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.

[4] M. Xie, S. Li, A. O. Glova, J. Hu and Y. Xie, "Securing Emerging Nonvolatile Main Memory With Fast and Energy-Efficient AES In-Memory Implementation," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2443-2455, Nov. 2018.

[5] D. Bui, D. Puschini, S. Bacles-Min, E. Beigné and X. Tran, "AES Datapath Optimization Strategies for Low-Power Low-Energy Multisecurity-Level Internet-of-Things Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 12, pp. 3281-3290, Dec. 2017.

[6] Z. Wang, T. Arslan and A. Erdogan, "Implementation of Hardware Encryption Engine for Wireless Communication on a Reconfigurable Instruction Cell Architecture," 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008), Hong Kong, 2008, pp. 148-152.

[7] I. Hammad, K. El-Sankary and E. El-Masry, "High-Speed AES Encryptor With Efficient Merging Techniques," in IEEE Embedded Systems Letters, vol. 2, no. 3, pp. 67-71, Sept. 2010.

[8] Yulin Zhang and Xinggang Wang, "Pipelined implementation of AES encryption based on FPGA," 2010 IEEE International Conference on Information Theory and Information Security, Beijing, 2010, pp. 170-173.

[9] A. P. A. Naidu and P. K. Joshi, "FPGA implementation of fully pipelined Advanced Encryption Standard," 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 2015, pp. 0649-0653.

[10] Q. Zhang and Q. Ding, "Digital Image Encryption Based on Advanced Encryption Standard (AES)," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, 2015, pp. 1218-1221.

[11] S. Shivkumar and G. Umamaheswari, "Performance Comparison of Advanced Encryption Standard (AES) and AES Key Dependent S-Box - Simulation Using MATLAB," 2011 International Conference on Process Automation, Control and Computing, Coimbatore, 2011, pp. 1-6.

[12] X. Wang, L. Han, C. Wang and X. Liu, "Based MATLAB on Advanced Encryption Standard (AES) IP Validation," 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, 2008, pp. 1329-1331.

[13] K. R. Kashwan and K. A. Dattathreya, "Improved serial 2D-DWT processor for advanced encryption standard," 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, 2015, pp. 209-213

[14] V. Hoang, V. Nguyen, A. Nguyen and C. Pham, "A low power AES-GCM authenticated encryption core in 65nm SOTB CMOS process," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, 2017, pp. 112-115.