JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

VLSI Implementation of Area and Delay Efficient Nano-AES for Internet-of-Things Application

[1]Archna Jyoti, [2]Shriddha Shrivastava
[1]Research Scholar, [2]Professor
Department of Electronic & Communication Engineering,
Lakshmi Narain College of Technology, Bhopal, India

Abstract : Cryptography techniques are considered 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 Nano-AES or lightweight cryptography with 256 bit key for IOT application. 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 Nano-AES lightweight cryptography gives better performance than previous in term of delay, throughput, transmission time, efficiency rate.

Index Terms **– IOT, Wireless, Security, Nano-AES lightweight Cryptography, Encryption, Decryption, Block Cipher, Simulation, Synthesis, Xilinx.**

I. INTRODUCTION

Nano-AES is a lightweight implementation of the Advanced Encryption Standard (AES) algorithm. It is designed for resource-constrained devices, such as those used in the Internet of Things (IoT). Nano-AES is optimized for area and power consumption, while still providing a high level of security. The Nano-AES algorithm is based on the original AES algorithm, but it has been simplified and optimized for smaller devices.

Cryptography in recent years with the advancement of VLSI has led to its implementation of Encryption and Decryption techniques, where the process of translating and converting plaintext into cipher text and vice versa is made possible.

The Nano-AES lightweight Encryption Algorithm (NA-LEA) is a 128-bit block cipher developed by South Korea in 2013 to provide confidentiality in high-speed environments such as big data and cloud computing, as well as Nano-AES lightweight environments such as IoT devices and mobile devices. LEA has three different key lengths: 128, 192, and 256 bits. LEA encrypts data about 1.5 to 2 times faster than AES, the most widely used block cipher in various software environments.

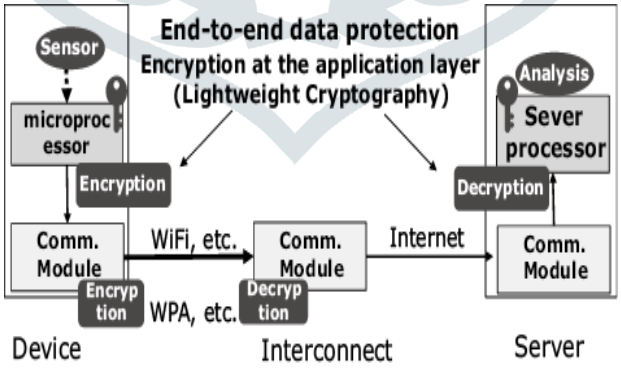

Figure 1: Example of Nano-AES lightweight cryptography applications

LEA is one of the cryptographic algorithms approved by the Korean Cryptographic Module Validation Program (KCMVP) and is the national standard of Republic of Korea (KS X 3246).

The block cipher LEA consisting of ARX operations, bitwise Rotation, and bitwise XOR for 32-bit words processes data blocks of 128 bits and has three different key lengths: 128, 192, and 256 bits. LEA with a 128-bit key, LEA with a 192-bit key, and LEA with a 256-bit key are referred to as “LEA-128”, “LEA-192”, and “LEA-256”, respectively. The number of rounds is 24 for LEA-128, 28 for LEA-192, and 32 for LEA-256. LEA has very good performance in a general-purpose software environment. In particular, it is possible to encrypt at a rate of about 1.5 to 2 times on average, compared to AES, the most widely used block cipher in various software environments. The tables below compare the performance of LEA and AES using FELICS (Fair Evaluation of Nano-AES lightweight Cryptographic Systems),[3] a benchmarking framework for evaluation of software implementations of

Nano-AES lightweight cryptographic primitives. Encryption is already applied as standard on the data link layer of communication systems such as the cellphone. Even in such a case, encryption in the application layer is effective in providing end-to-end data protection from the device to the server and to ensure security independently from the communication system (Fig. 1). Then encryption must be applied at the processor processing the application and on unused resources and hence should desirably be as Nano-AES lightweight as possible.

The biggest security-related threat of IoT systems from the traditional IT systems is that even using devices for data collection from the real world can become a target of cyberattacks. For example, the purpose of applying IoT to a plant is to significantly improve the productivity and maintainability by collecting data from a large number of sensors installed in production equipment, by analyzing it and performing autonomous control in real time. If sensor data should be falsified during this process, incorrect analysis results would be induced and erroneous control would result due to such an occurrence having the potential of leading to major damage. Moreover, since measurement data and control commands are trade secrets associated with the know-how of production and management, preventing leakages is also important from the viewpoint of competitiveness. Even if there is no problem at present, it is necessary to consider the effect of threats that might become evident in the future.

II. PROPOSED METHODOLOGY

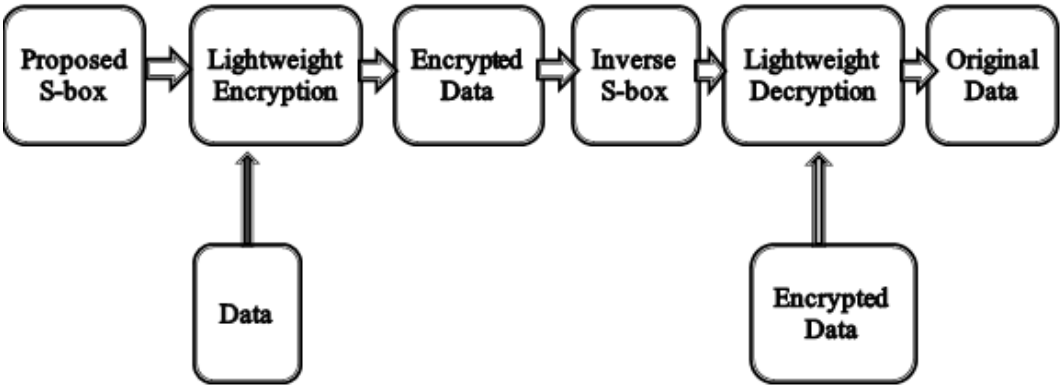

Figure 2: Flow Chart

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.

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.

III. SIMULATION AND RESULT

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.

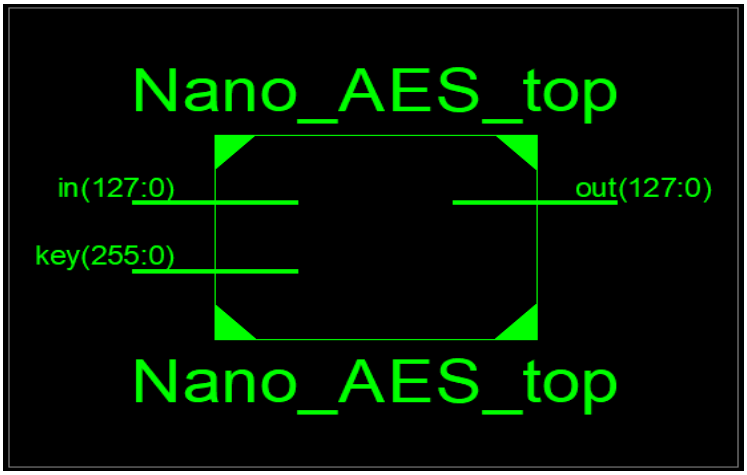

Figure 3: Top View

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

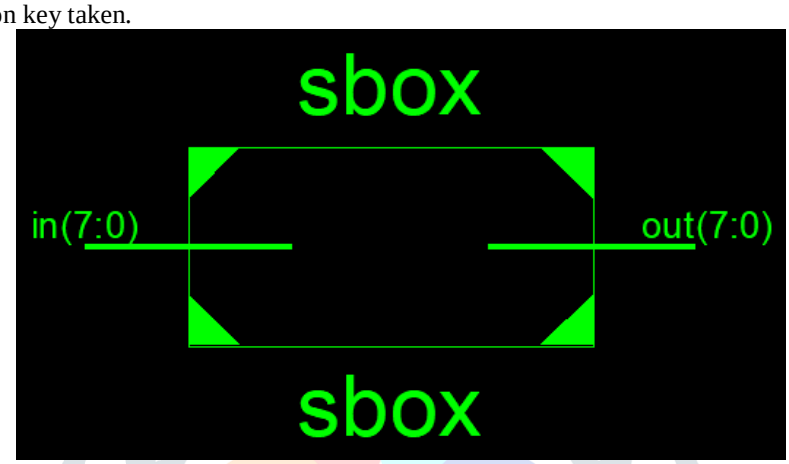

Figure 4: Top module of proposed 1D S-box

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.

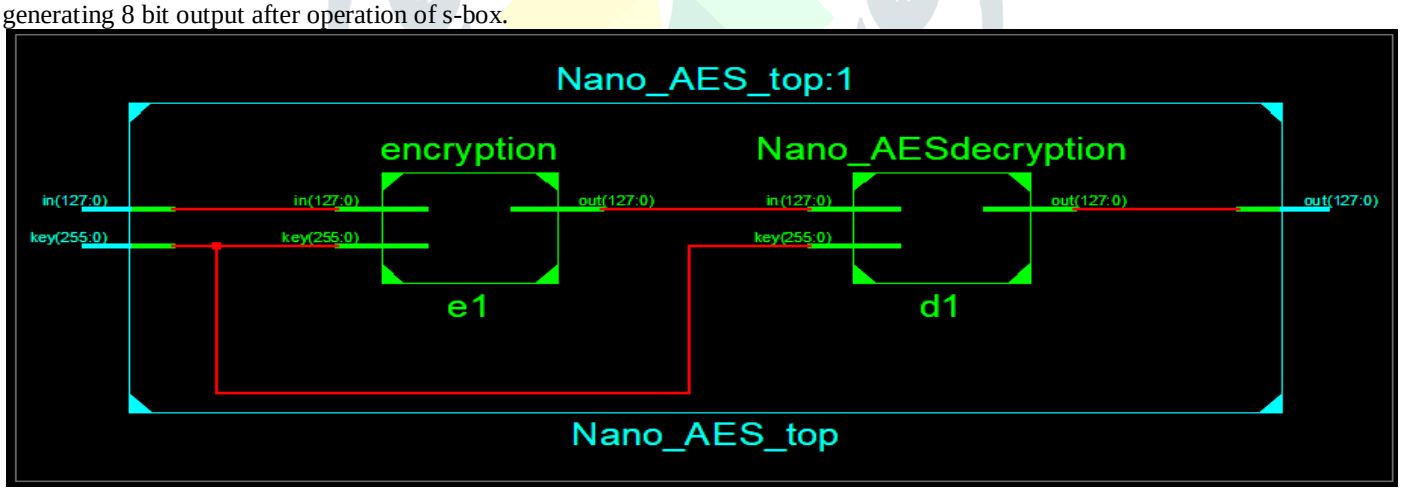

Figure 5: RTL view of Encryption and Decryption Process

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.

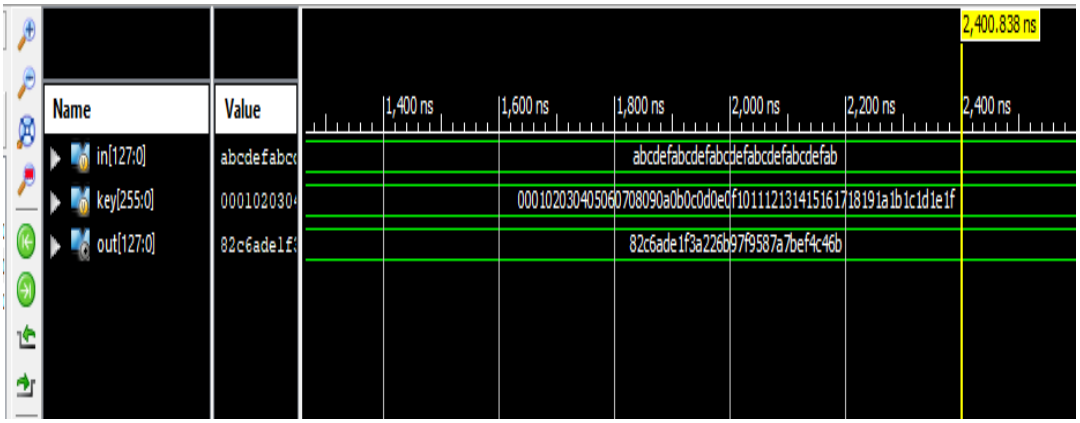

Figure 6: Encryption process

Figure 6 presents the encryption process of the proposed Nano-AES lightweight cryptography algorithm.

Input – abcdefabcdefabcdefabcdefabcdefab

Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

Output - 82c6ade1f3a226b97f9587a7bef4c46b

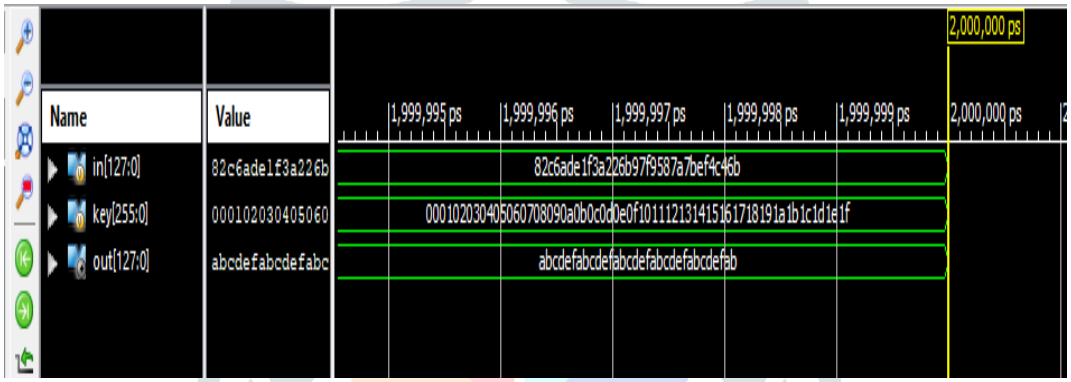

Figure 7: Decryption Process

Figure 7 presents the decryption process of the proposed Nano-AES lightweight cryptography algorithm.

Input – 82c6ade1f3a226b97f9587a7bef4c46b

Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

Output - abcdefabcdefabcdefabcdefabcdefab

Table 1: Result Comparison

Sr No.	Parameters	Previous Result [1]	Proposed Result
1	S- box components	86	16
2	Combinational components	19,120	13016
3	S box delay (ns)	6.830	3.12
4	Combinational delay (ns)	20	13.49
5	Plain text	64 bit	128 bit
6	Frequency	200 MHz	450 MHz
7	Throughput	250 Mbps	1 Gbps

IV. CONCLUSION

This paper presents the implementation of data security algorithm based on Nano-AES lightweight cryptography 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 Nano-AES lightweight cryptography 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 Nano-AES lightweight cryptography delay is 20ns by previous and while proposed it is 13.49ns. The frequency and throughput is 450 MHz and 1 Gbps in proposed and 200 MHz and 250 Mbps in previous. Therefore the simulation results achieved significant better performance than the existing research work.

REFERENCES

1. K. Shahbazi and S. -B. Ko, "Area-Efficient Nano-AES Implementation for Internet-of-Things Devices," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 1, pp. 136-148, Jan. 2021, doi: 10.1109/TVLSI.2020.3033928.
2. 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.
3. 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), Guwahati, India, 2020, pp. 115-121, doi: 10.1109/ISEA-ISAP49340.2020.235009.
4. 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.
5. B. 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.
6. Y. Wang and Y. Ha, "FPGA-Based 40.9-Gbits/s Masked AES With Area Optimization for Storage Area Network," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 1, pp. 36-40, Jan. 2013.
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. S. H. Kamali, R. Shakerian, M. Hedayati and M. Rahmani, "A new modified version of Advanced Encryption Standard based algorithm for image encryption," 2010 International Conference on Electronics and Information Engineering, Kyoto, 2010, pp. V1-141-V1-145.
9. 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.
10. A. 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.
11. S. Dahiya and M. Bohra, "Hybrid parallel partial model for robust & secure authentication in healthcare IoT environments," 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), Mathura, 2017, pp. 239-243.
12. G. Singh Rajput, R. Thakur, R. Tiwari, “VLSI implementation of lightweight cryptography technique for FPGA-IOT application”, Materials Today: Proceedings, 2023, ISSN 2214-7853, //doi.org/10.1016/j.matpr.2023.03.486.