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 Architecture of Lightweight Cipher Cryptography Technique for High Speed FPGA Applications

[1]Sagar, [2]Dr. Bharti Chourasia

[1]Research Scholar, [2]Associate Professor & HOD
Department of Electronics and Communication Engineering,
RKDF Institute of Science & Technology, SRK University, 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 VLSI architecture of lightweight cipher cryptography technique for high speed FPGA Applications. 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 lightweight cryptography gives better performance than previous in term of delay, throughput, transmission time, efficiency rate.

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

I. INTRODUCTION

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 Lightweight Encryption Algorithm (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 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.

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 Lightweight Cryptographic Systems),[3] a benchmarking framework for evaluation of software implementations of lightweight cryptographic primitives.

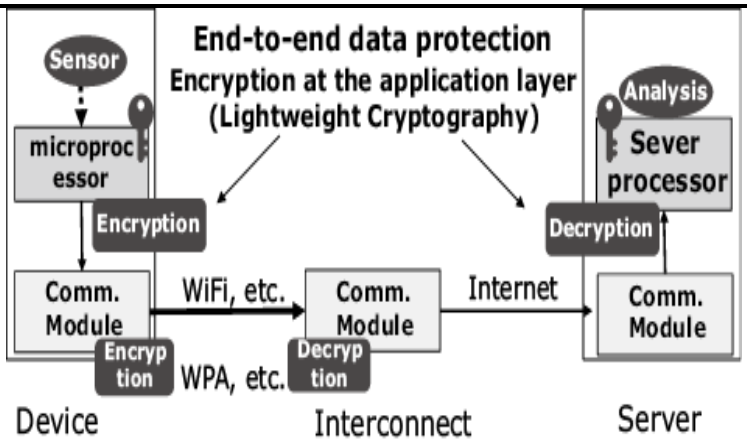

Figure 1: Example of lightweight cryptography applications

Encryption is already applied as standard on the data link layer of communication systems such as the cell phone. 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 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 cyber attacks. 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

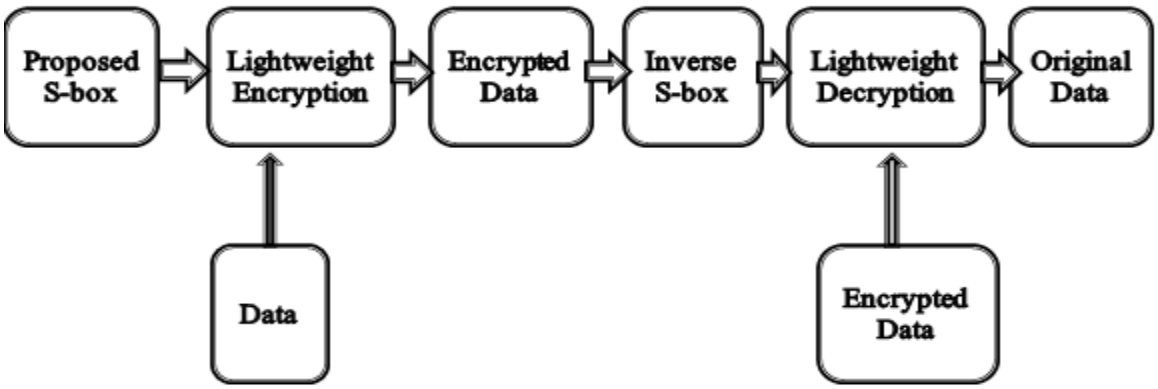

Figure 2: Flow Chart

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.

In this paper, we have implemented WiMax/IOT Security using Lightweight cryptography 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.

III. RESULT AND ANALYSIS

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.

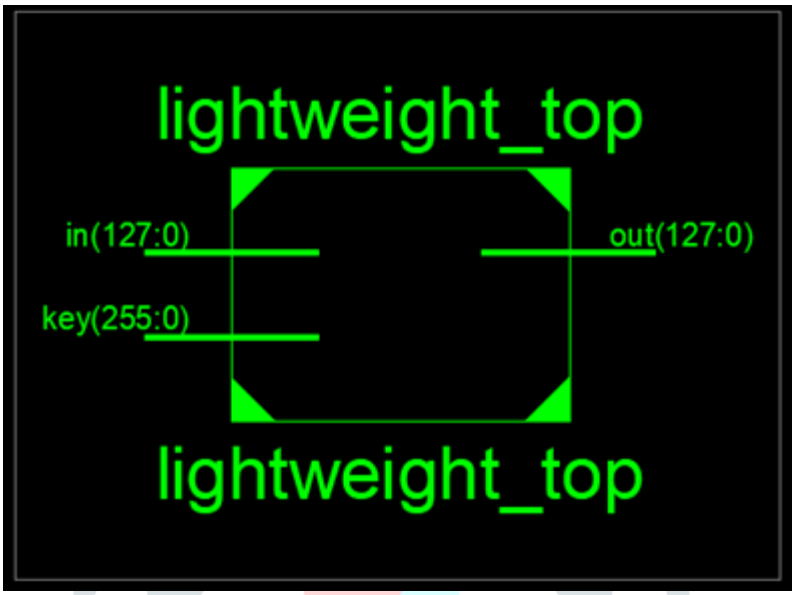

Figure 3: Top View

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

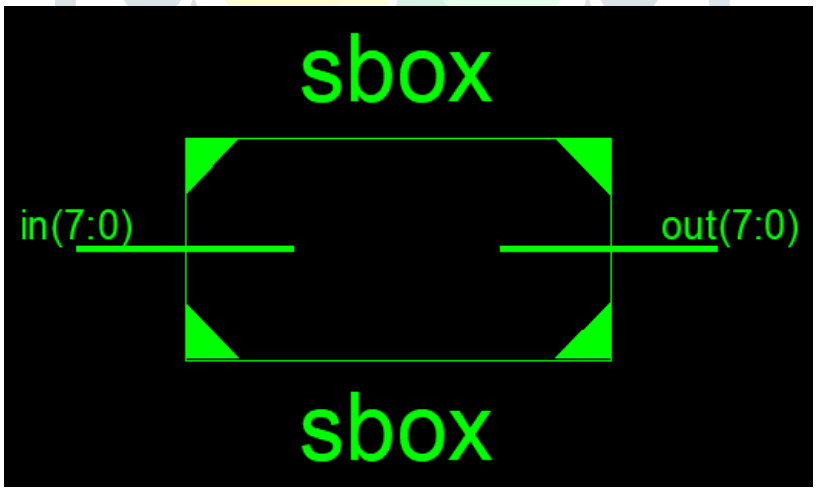

Figure 4: Top module of proposed 1D S-box

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.

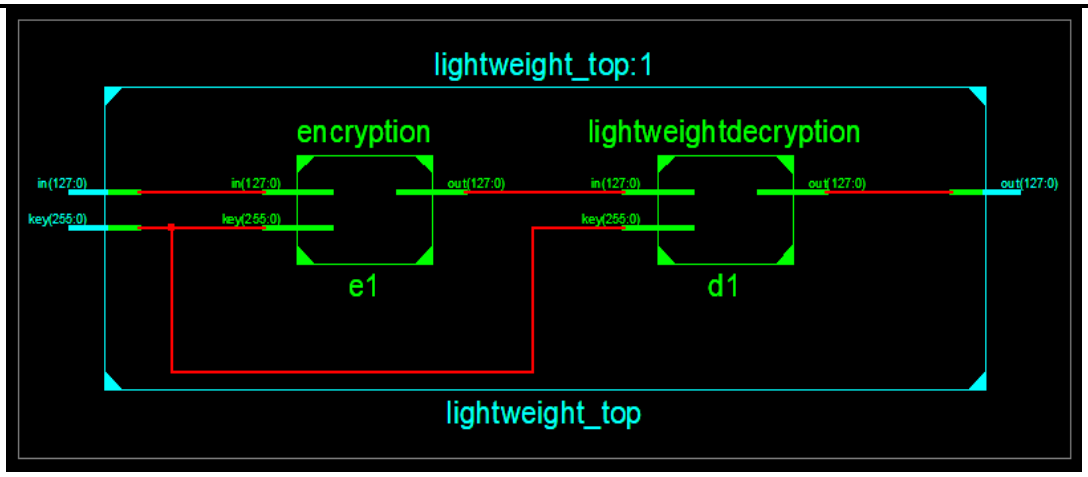

Figure 5: RTL view of Encryption and Decryption Process

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.

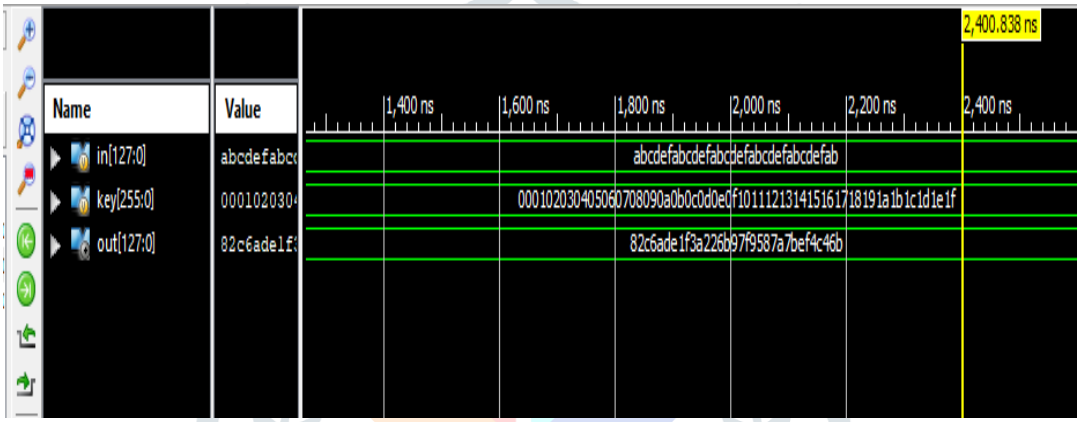

Figure 6: Encryption process

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

Input – abcdefabcdefabcdefabcdefabcdefab
Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Output - 82c6ade1f3a226b97f9587a7bef4c46b

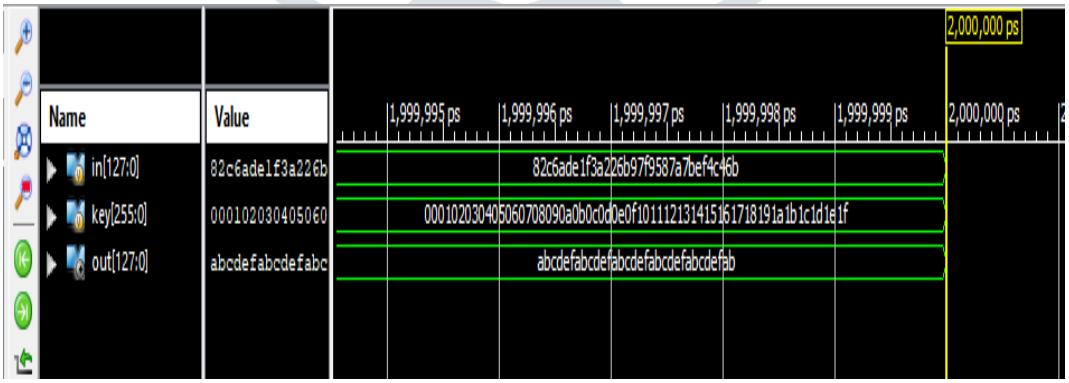

Figure 7: Decryption Process

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

Input – 82c6ade1f3a226b97f9587a7bef4c46b
Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Output - abcdefabcdefabcdefabcdefabcdefab

Table 1: Result Comparison

Sr No.	Parameters	Previous Work	Proposed Work
1	Input bit	80	128
2	Frequency	10 MHz	23 MHz

3	Area	28860.580	13017
4	Total Power	0.2535 mW	0.18mW
5	Throughput	250 Mbps	2949 Mbps
6	Delay or Latency	100 ns	43.398ns

IV. CONCLUSION

This paper presents implementation of lightweight crypto VLSI architecture for high speed FPGA applications. Simulation is conducted using the Xilinx ISE 14.7 software utilizing the verilog code. The simulation results reveals that the prior work is based on the 80 bit data input while suggested work is based on the 128 bit data input with 256 bit key so that it give higher security. The value of frequency is 23 MHz in the proposed work whereas 10 MHz in the preceding work. The overall throughput is 2949 Mbps in the proposed work whereas 250 Mbps in the existing and the total latency is 43.39ns in the proposed and 100ns in the previous work.

REFERENCES

1. J. G. Pandey, A. Laddha and S. D. Samaddar, "A Lightweight VLSI Architecture for RECTANGLE Cipher and its Implementation on an FPGA," 2020 24th International Symposium on VLSI Design and Test (VDAT), 2020, pp. 1-6, doi: 10.1109/VDAT50263.2020.9190623.
2. P. B.S, N. K.J and N. J. C.M, "MEC S-box based PRESENT Lightweight Cipher for Enhanced Security and Throughput," 2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 2020, pp. 212-217, doi: 10.1109/DISCOVER50404.2020.9278038.
3. B. Hajri, M. M. Mansour, A. Chehab and H. Aziza, "A Lightweight Reconfigurable RRAM-based PUF for Highly Secure Applications," 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2020, pp. 1-4, doi: 10.1109/DFT50435.2020.9250829.
4. B. Richter and A. Moradi, "Lightweight Ciphers on a 65 nm ASIC A Comparative Study on Energy Consumption," 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2020, pp. 530-535, doi: 10.1109/ISVLSI49217.2020.000-2.
5. P. Singh, B. Acharya and R. K. Chaurasiya, "Efficient VLSI Architectures of LILLIPUT Block Cipher for Resource-constrained RFID Devices," 2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2019, pp. 1-6, doi: 10.1109/CONECCT47791.2019.9012869.
6. R. Sadhukhan, N. Datta and D. Mukhopadhyay, "Power Efficiency of S-Boxes: From a Machine-Learning-Based Tool to a Deterministic Model," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 12, pp. 2829-2841, Dec. 2019, doi: 10.1109/TVLSI.2019.2925421.
7. T. Chen, K. Hou, W. Beh and A. Wu, "Low-Complexity Compressed-Sensing-Based Watermark Cryptosystem and Circuits Implementation for Wireless Sensor Networks," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2485-2497, Nov. 2019, doi: 10.1109/TVLSI.2019.2933722.
8. M. Zhang, L. Zhang, L. Jiang, F. T. Chong and Z. Liu, "Quick-and-Dirty: An Architecture for High-Performance Temporary Short Writes in MLC PCM," in IEEE Transactions on Computers, vol. 68, no. 9, pp. 1365-1375, 1 Sept. 2019, doi: 10.1109/TC.2019.2900036.
9. M. M. Wong, V. Pudi and A. Chattopadhyay, "Lightweight and High Performance SHA-256 using Architectural Folding and 4-2 Adder Compressor," 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2018, pp. 95-100, doi: 10.1109/VLSI-SoC.2018.8644825.
10. S. Mandal, D. Bhattacharjee, Y. Tavva and A. Chattopadhyay, "ReRAM-based In-Memory Computation of Galois Field arithmetic," 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2018, pp. 1-6, doi: 10.1109/VLSI-SoC.2018.8644772.
11. J. G. Pandey, T. Goel, M. Nayak, C. Mitharwal, A. Karmakar and R. Singh, "A High-Performance VLSI Architecture of the Present Cipher and its Implementations for SoCs," 2018 31st IEEE International System-on-Chip Conference (SOCC), 2018, pp. 96-101, doi: 10.1109/SOCC.2018.8618487.
12. T. Goel, J. G. Pandey and A. Karmakar, "A High-Performance and Area-Efficient VLSI Architecture for the PRESENT Lightweight Cipher," 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), 2018, pp. 392-397, doi: 10.1109/VLSID.2018.96.
13. V. B. Y. Kumar, D. Shah, M. Datar and S. B. Patkar, "Lightweight Forth Programmable NoCs," 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), 2018, pp. 368-373, doi: 10.1109/VLSID.2018.92.
14. A. Villegas, R. Asenjo, A. Navarro, O. Plata and D. Kaeli, "Lightweight Hardware Transactional Memory for GPU Scratchpad Memory," in IEEE Transactions on Computers, vol. 67, no. 6, pp. 816-829, 1 June 2018, doi: 10.1109/TC.2017.2776908.