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

FPGA Implementation of Modified Cryptography for VLSI Security Applications

[1]Jyoti Soni, [2]Dr. M Zahid Alam
[1]Research Scholar, [2]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. The security is major concern as developing the application under 5G-IOT. FPGA technology is upgrading to enhance the performance of latency and area. Many of the security algorithm is already developed but its less perform under advance FPGA-IOT constraints. The advance encryption standard is one of the best security algorithm and using in real time high secure application but its need to modify for FPGA-IOT applications. This paper proposes FPGA implementation of modified cryptography for VLSI security Applications. Modified-AES algorithm is a fast lightweight encryption algorithm for advance security with low latency. Simulation is performed using the Xilinx ISE 14.7 Software with verilog language. The simulation results are verified using Isim simulator in Xilinx test bench.

***IndexTerms* –** Cryptography, Encryption, Decryption, Xilinx ISE, Verilog, VLSI, FPGA, IOT, 5th G, Security.

I. INTRODUCTION

AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to a form called ciphertext. Decryption of the ciphertext converts the data back into its original form, which is called plaintext.

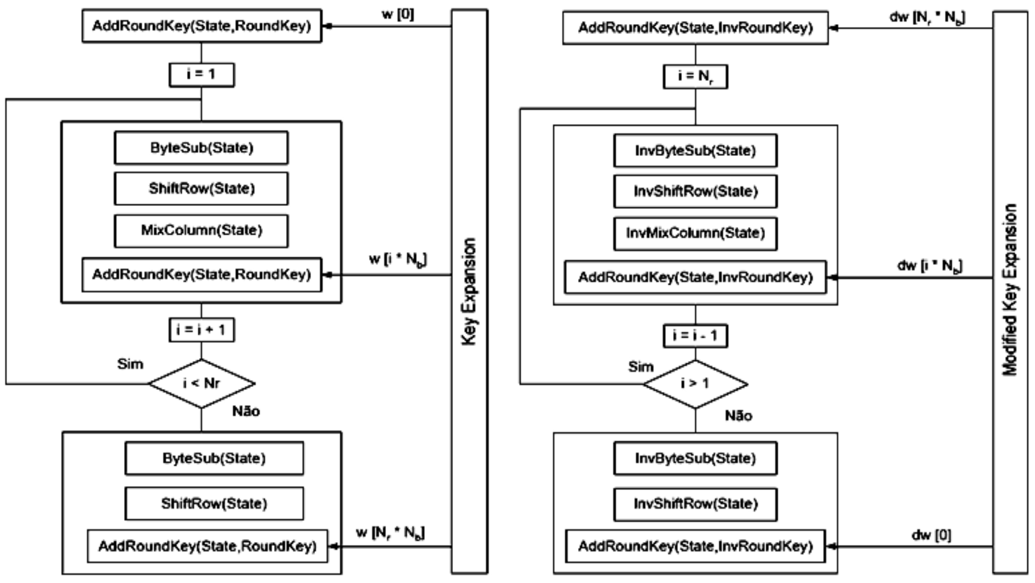


Figure 1: Rinjdael AES Encryption and Decryption Process

The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of bits. Compared to software implementations, hardware implementations of the AES algorithm provide more physical security as well as higher speed. The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. Key Expansion generates a key Schedule that is used in Cipher and Inverse Cipher procedure.

AES encryption is an efficient scheme for both hardware and software implementation. As compare to software implementation, hardware implementation provides greater physical security and higher speed. Hardware implementation is useful in wireless security like military communication and mobile telephony where there is a greater emphasis on the speed of communication. Most of the work has been presented on hardware implementation of AES using FPGA.

II. PROPOSED APPROACH

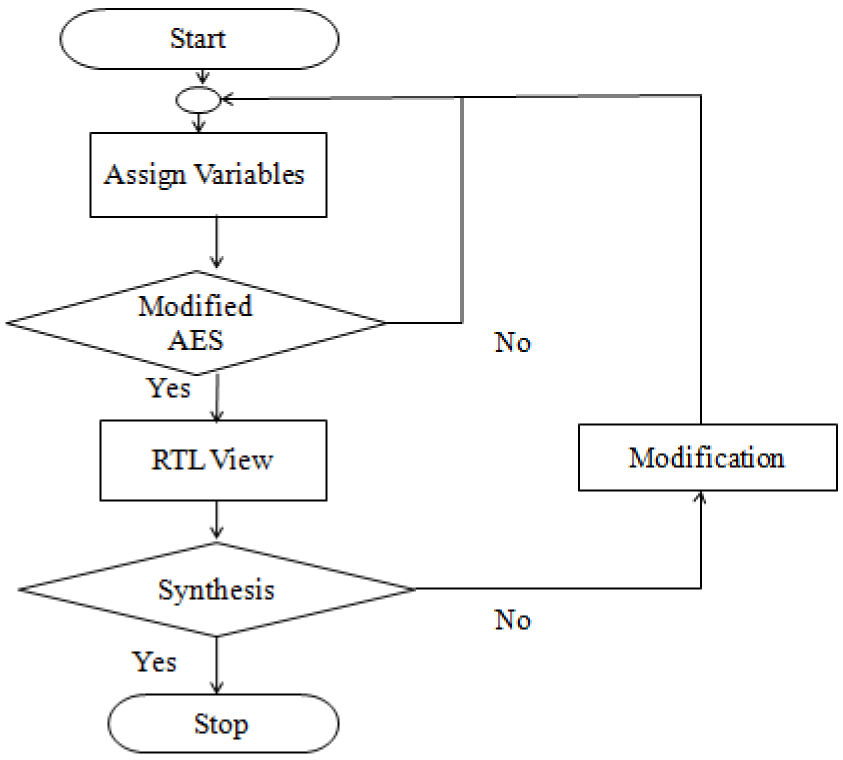


Figure 2: Flow Chart

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 RESULTS

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, which is performed in VLSI Xilinx ISE 14.7 with verilog coding.

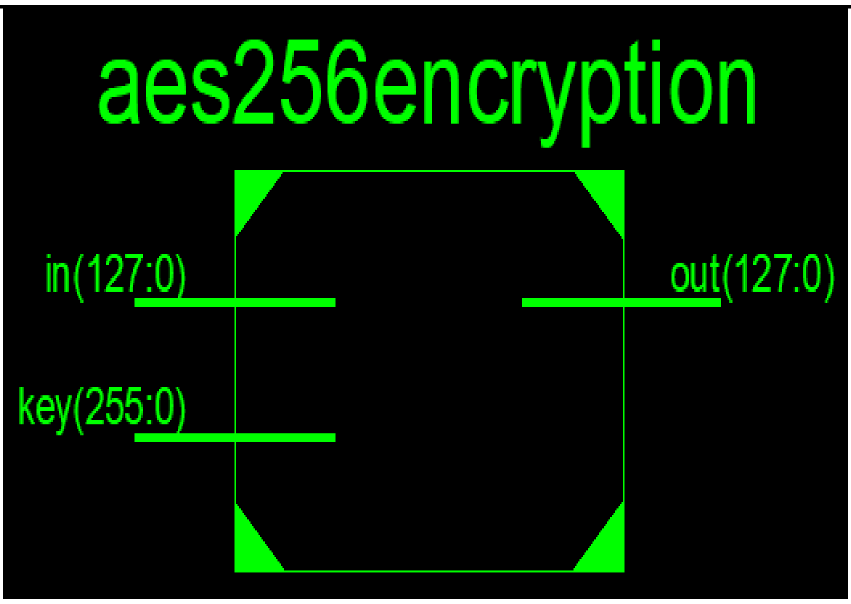


Figure 3: RTL Schematics of WiMax/IOT MAES

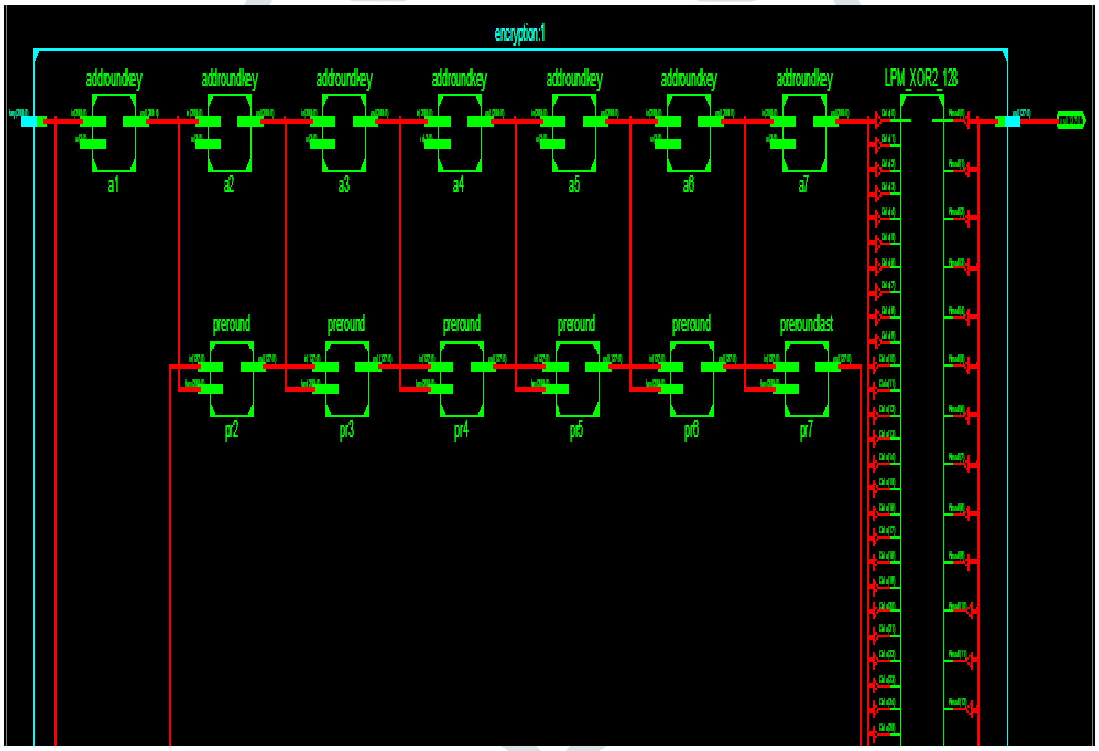
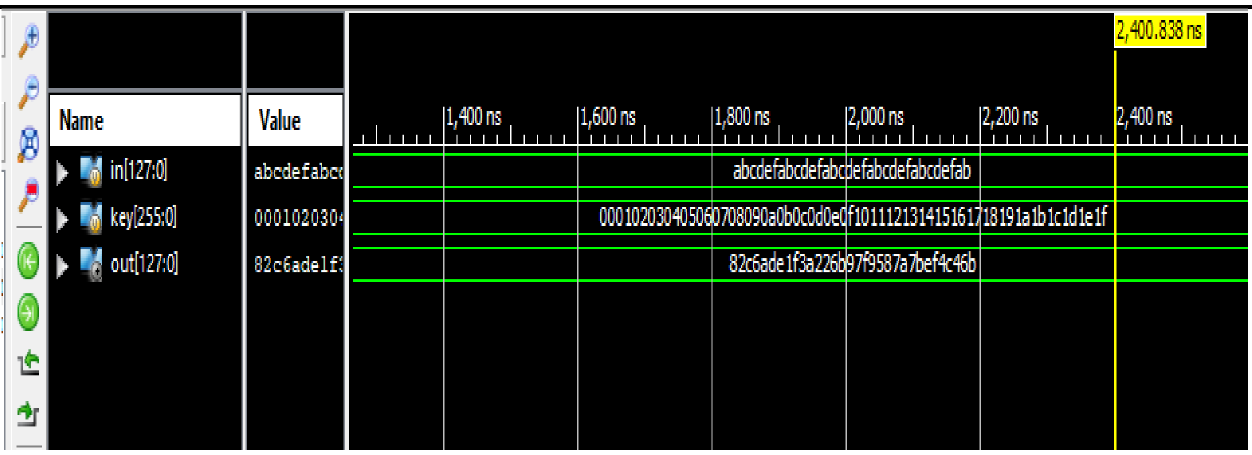


Figure 4: RTL view of proposed MAES algorithm

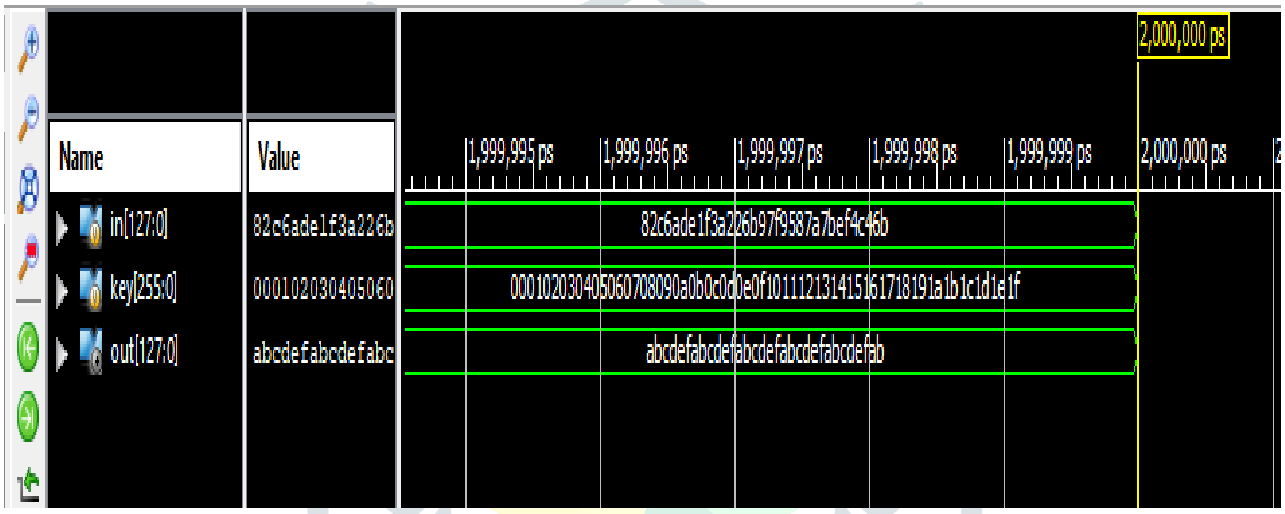
It is designed WiMax/IOT Security using Advanced Encryption Standard Cryptographic Algorithm. We have used Verilog for this purpose. We have used Xilinx ISE which has given synthesis results, as summarized in Table 1. Below. Also, we have depicted the pictorial representation of the results, which is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



```
Name          Value
in[127:0]     abcdefabcdefabcdefabcdefabcdefab
key[255:0]    000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
out[127:0]    82c6ade1f3a226b97f9587a7bef4c46b
```

Figure 5: Encryption Process

In figure 5, presents the encryption process of the proposed modified AES algorithm.
Input – abcdefabcdefabcdefabcdefabcdefab
Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Output - 82c6ade1f3a226b97f9587a7bef4c46b
.



```
Name          Value
in[127:0]     82c6ade1f3a226b97f9587a7bef4c46b
key[255:0]    000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
out[127:0]    abcdefabcdefabcdefabcdefabcdefab
```

Figure 6: Decryption Process

In figure 6, presents the decryption process of the proposed modified AES algorithm.
Input – 82c6ade1f3a226b97f9587a7bef4c46b
Key-h000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Output - abcdefabcdefabcdefabcdefabcdefab

Table 1: Result Comparison

Sr No.	Parameters	Previous Result [1]	Proposed Result
1	FPGA	Virtex-7	Virtex-7
2	Area	3298	1301
3	Frequency	364.26MHz	794 MHz
4	Throughput	16513.12 Mbps	101632 Mbps

These comparison tables present various parameters values of previous work and proposed 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.

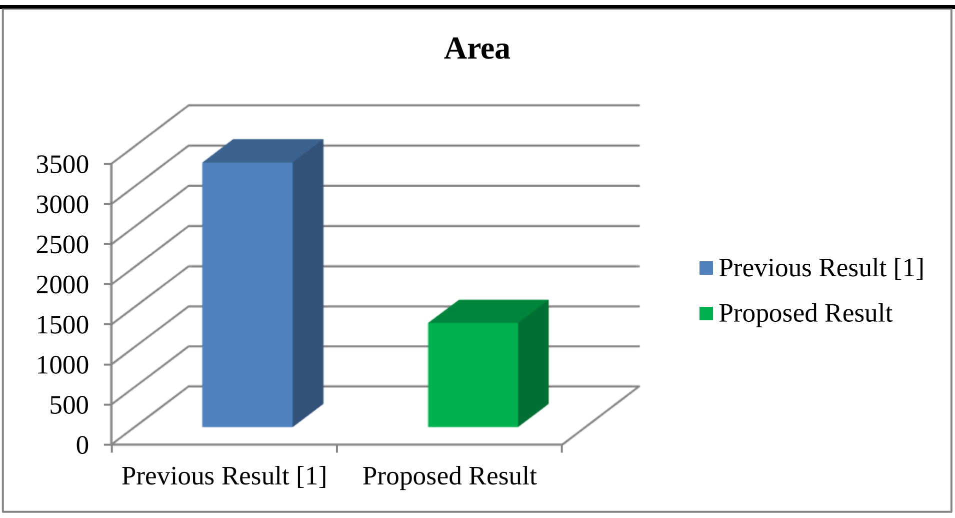


Figure 7: Area

Figure 7 is showing graphical representation of latency (delay). Therefore it is clear that, proposed MAES give better performance than existing AES.

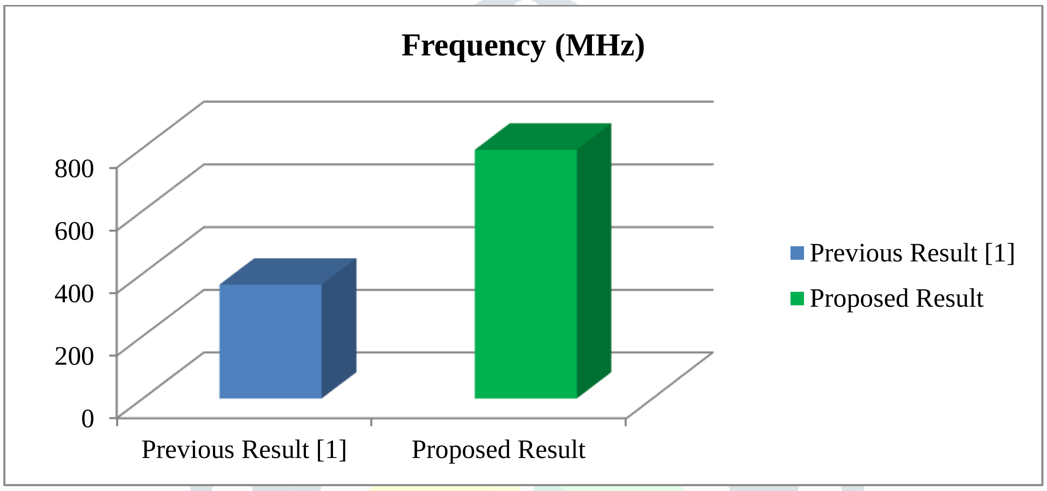


Figure 8: Frequency

Figure 7 & 8 showing graphical representation of area and frequency comparison. Therefore it is clear that proposed MAES gives better performance than previous MAES.

IV. CONCLUSION

This paper presents VLSI architecture of modified AES with 256-bit key for high speed FPGA-IOT Applications. Proposed algorithm 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). A lightweight version of advanced encryption standard (AES) meets the requirements of FPGA-IOT applications. A Modified AES is proposed by formulating a novel equation for constructing a square matrix in affine transformation phase of MAES.

REFERENCES

1. D. e.S. Kundi, A. Khalid, A. Aziz, C. Wang, M. O'Neill and W. Liu, "Resource-Shared Crypto-Coprocessor of AES Enc/Dec With SHA-3," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 12, pp. 4869-4882, Dec. 2020, doi: 10.1109/TCSI.2020.2997916.
2. S. Al-Shara'a, R. K. Ibraheem and O. Bayat, "Implementation of cryptanalysis based on FPGA hardware using AES with SHA-1," 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), 2019, pp. 1-7, doi: 10.1109/SmartNets48225.2019.9069786.
3. A. S. Dewi and H. Setiawan, "Implementation of SHA-256 and AES-256 for Securing Digital Al Quran Verification System," 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019, pp. 1-8, doi: 10.1109/ICIC47613.2019.8985960.
4. P. A. Wibowo Putro and D. Arumsari, "Designing and Building Disposition- EL Application by Applying AES-256 and RSA-2048," 2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), 2019, pp. 163-168, doi: 10.1109/ICIMCIS48181.2019.8985232.
5. N. H. Hussein, "Cloud-Based Efficient and Secure Scheme for Medical Images Storage and Sharing using ECC and SHA-3," 2019 2nd Scientific Conference of Computer Sciences (SCCS), 2019, pp. 109-115, doi: 10.1109/SCCS.2019.8852620.

6. T. N. Dang and H. M. Vo, "Advanced AES Algorithm Using Dynamic Key in the Internet of Things System," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), 2019, pp. 682-686, doi: 10.1109/CCOMS.2019.8821647.
7. N. A. Fauziah, E. H. Rachmawanto, D. R. I. Moses Setiadi and C. A. Sari, "Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application," 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018, pp. 146-151, doi: 10.1109/ISRITI.2018.8864485.
8. G. C. Prasetyadi, A. Benny Mutiara and R. Refianti, "File encryption and hiding application based on advanced encryption standard (AES) and append insertion steganography method," 2017 Second International Conference on Informatics and Computing (ICIC), 2017, pp. 1-5, doi: 10.1109/IAC.2017.8280584.
9. O. C. Briliyant and A. Baihaqi, "Implementation of RSA 2048-bit and AES 128-bit for Secure e-learning web-based application," 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017, pp. 1-5, doi: 10.1109/TSSA.2017.8272903.
10. K. D. B. Utama, Q. M. R. Al-Ghazali, L. I. B. Mahendra and G. F. Shidik, "Digital signature using MAC address based AES-128 and SHA-2 256-bit," 2017 International Seminar on Application for Technology of Information and Communication (iSemantic), 2017, pp. 72-78, doi: 10.1109/ISEMANTIC.2017.8251846.
11. T. Hermawan and R. W. Wardhani, "Implementation AES with digital signature for secure web-based electronic archive," 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), 2016, pp. 1-6, doi: 10.1109/ICITEED.2016.7863268.
12. N. Kate and J. V. Katti, "Security of remote voting system based on Visual Cryptography and SHA," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016, pp. 1-6, doi: 10.1109/ICCUBEA.2016.7860071.
13. Z. Kouser, M. Singhal and A. M. Joshi, “FPGA implementation of advanced Encryption Standard algorithm,” *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, 2016, pp. 1-5.
14. N. D. Vaidya, Y. A. Suryawanshi and M. Chavan, “Design for enhancing the performance of Advance Encryption Standard algorithm VHDL,” *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, 2016, pp. 1-5.