International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

FPGA Based Cryptographic Hardware for Embedded Systems

Ravi Kumar M[1]*, Dr. Girish H*[2]*, Anusha Dixith G*[3]*, Aina Saba N*[4]*, Anusha K V*[5]*, Chandana M*[6]

[1]Assistant Professor, Department of Electronics and Communication, Cambridge Institute of Technology (CITech), KR Puram, Bengaluru, India.
[2]Professor, Department of Electronics and Communication, Cambridge Institute of Technology (CITech), KR Puram, Bengaluru, India.
[3,4,5,6] Student, Department of Electronics and Communication, CITech, KR Puram, Bengaluru, India.

ABSTRACT:

In most applications, such as online banking, e-commerce, military, satellite, wireless communications, electronic devices, digital image processing, etc., hardware security is crucial. The process of transforming regular, simple language into incomprehensible text and vice versa is known as cryptography. Public key cryptography, hash functions, and symmetric key cryptography are the three categories of cryptographic techniques. The Advanced Encryption Standard (AES) and Data Encryption Standard are two examples of symmetric key algorithms that employ the same key for both encryption and decryption. It uses less computing resources, is simpler to implement, and is substantially faster. The suggested 256-bit AES algorithm has excellent area and power optimization in the key schedule and sub byte blocks. The S-box block has been reused in order to optimize. We are making the most of the algorithm with a new approach where internal operations are 32-bit operations, as compared to 128-bit operations. The proposed implementation helps in re-using the same hardware in a pipelined fashion which results in an area reduction using slice registers, slice LUT's and LUT-FF Pairs. This in turn results in a power reduction by a FPGA implementation. The throughput (Mbps) of the proposed implementation using BASYS FPGA improved.

Keywords— AES (Advanced Encryption Standard), FPGA (field programmable gate array), LUT (Look up table), Mbps (megabit per second), sub (sub bytes), shift (shift rows), mix (mix column), add (add round key).

1. Introduction

The Advanced Encryption Standard (AES), a symmetric key block which is published by the National Institute of Standards and Technology (NIST) in December 2001. It is a non-Feistel block cipher that encrypts and decrypts a fixed data block of 128-bits. There are three different key lengths. The encryption/decryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

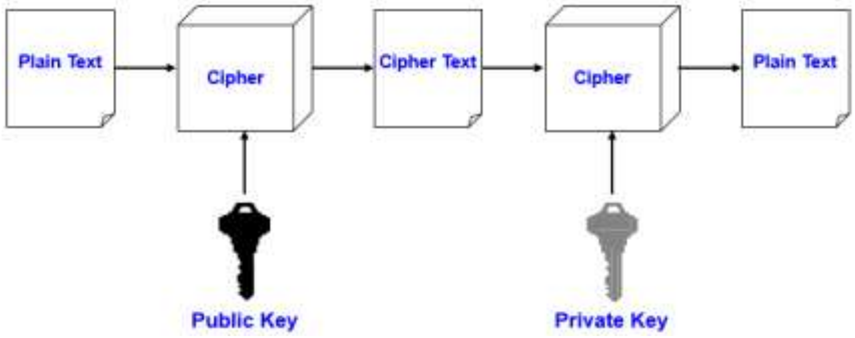
It has two main Components:

- ***Encryption:***

Practice of hiding messages so that they cannot be read by anyone other than the intended recipient

- ***Authentication & Integrity:***

Ensuring that users of data/resources are the persons they claim to be and that a message has not been surreptitiously altered



Symmetric algorithms:

Algorithms in which the key for encryption and decryption are the same are Symmetric

Example: Caesar Cipher

Types:

1. Block Ciphers
   - Encrypt data one block at a time (typically 64 bits, or 128 bits)
   - Used for a single message
2. Stream Ciphers
   - Encrypt data one bit or one byte at a time
   - Used if data is a constant stream of information

Substitution-boxes are used in contemporary block ciphers to convert plaintext data nonlinearly and produce cipher-text data that is sufficiently confusing. The strength and security of these block ciphers have been shown to be strongly dependent on the cryptographic strength of the underlying substitution-boxes. For the simple reason that only they are accountable for adding the necessary complexity and nonlinearity to the security system, which can annoy attackers. Consequently, several ideas have been investigated to build robust S-boxes. In this study, a revolutionary basic modular strategy is examined for the first time to create nonlinear S-box with the same goal in mind. Three operations make up the new modular approach: permutation, modular inverses, and new transformation. A number of highly nonlinear S-boxes can be easily constructed with slight changes in the novel transformation parameters. An example S-box is presented whose critical performance assessment against some benchmarking criterions such as high nonlinearity, absence of fixed points, fulfillment of SAC and BIC properties, low differential uniformity and linear approximation probability and comparison with recent S-boxes demonstrate its upright cryptographic potentiality. In addition, an image encryption algorithm is also proposed wherein the generated S-box is applied to perform the pixels shuffling and substitution for strong statistical and differential encryption performance.

It is simple to generate several extremely nonlinear S-boxes by making small adjustments to the unique transformation parameters. We present an example S-box whose upright cryptographic potential is demonstrated by comparison with recent S-boxes and critical performance assessment against some benchmarking criteria, such as high nonlinearity, absence of fixed points, fulfillment of SAC and BIC properties, low differential uniformity, and linear approximation probability. Furthermore, a proposal is made for an image encryption technique that utilizes the created S-box to execute pixel replacement and shuffling in order to achieve robust statistical and differential encryption performance.

Information and data Today's technology lifestyle includes communication as a crucial component, and it is regarded as one of an individual's or organization's most valuable assets. If information is compromised about its secrecy, then the information canbe used for harmful purposes. These block ciphers use two major operations of substitution and permutation for the transformation of data into a perplexing form.

II. METHODOLOGY

Figure 1 shows the architecture of our proposed system, built on the SoC FPGA platforms. In particular, we integrated the self-designed IP core to perform symmetric cryptography is AES 128-bit with the hard processor system (HPS), which consists of dual ARM core and SDRAM memory.

The architecture of our suggested system, which is based on SoC FPGA platforms, is depicted in Figure 1. Specifically, we linked the hard processor system (HPS)—which comprises of a dual ARM core and SDRAM memory—with the self-designed IP core to perform AES 128-bit symmetric cryptography.

To boost processing and computing power, the system can also incorporate additional unique hardware IP cores on the FPGA side. To control the entire system, we developed a special Linux kernel on the HPS side for the Linux operating system. In addition, we created unique drivers to interface with our hardware in order to create apps related to cryptography. Our cryptosystem has finally been integrated into the DE10 Standard board. Furthermore, our suggested solution offers IoT system operating models that can be used in practice. Specifically, the DE10-Standard board functions as a node or a gateway. Through the use of sensors, the node devices gather, process, and encrypt data from the surrounding environment—such as temperature, humidity, and other sorts of information—before sending it to the gateway devices via wireless networks. Gateway devices gather data, decrypt it for local processing, and then re-encrypt it before sending it across the Internet network to a server or cloud. In contrast, the node devices received commands from the server through the gateway devices, which then forwarded them to the actuators. Due to the fact that wireless networks and the Internet are public spaces, there are numerous concerns, including the possibility of data tampering, system hacking, and stolen information. Consequently, we employed a self-designed AES 128-bit IP core to convert data transmitted between the server and the gateway and between gateway and node devices to unintelligible form so that unauthorized persons cannot read it.

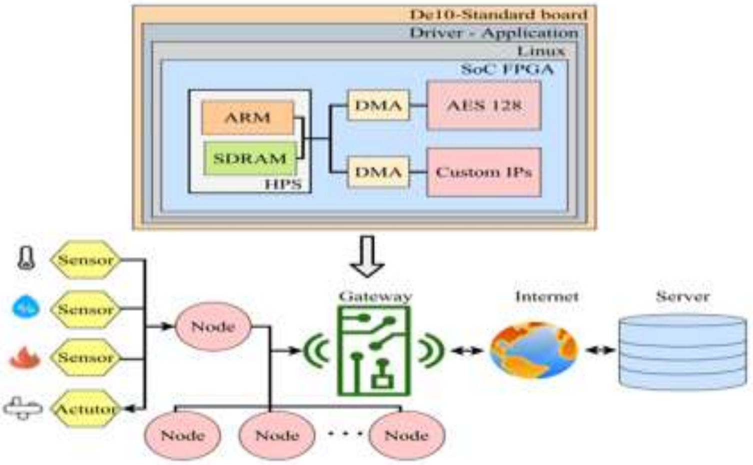
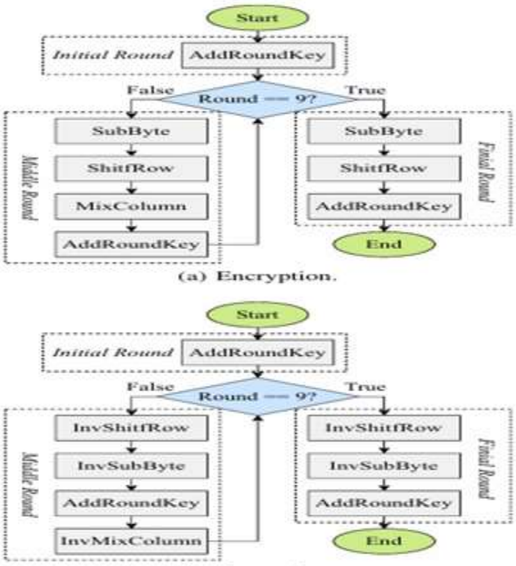


Figure 1: System architecture.

III. AES ALGORITHM



(a) Encryption.

(b) Decryption.

Figure 3: AES 128-bit process.

A cryptographic algorithm created for security is called the AES . In September 1997, a request was made by the National Institute of Standards and Technology (NIST) for the AES algorithm to take the role of the DES algorithm. After the 15 candidate algorithms were chosen, only five finalists were revealed in August 1999, a year later. These five algorithms are Twofish, Rijndael, MARS, RC6, and Serpent. In October 2000, the Rijndael algorithm—created by Joan Daemen and Vincent Rijmen—was chosen as the AES development process winner. Nov. 26, 2001 saw the release of Federal Information Processing Standards Publication 197 (FIPS 197), which describes the Advanced Encryption Standard (AES) algorithm. The three primary components of the AES algorithm are the Cipher, Inverse Cipher, and Key Expansion. A Key Schedule is produced by the Key Expansion and is utilized in the Cipher and Inverse Cipher process. The Inverse Cipher transforms cipher text back into the original data known as plaintext, whereas the Cipher transforms data into an unintelligible form known as cipher text.

a. Encryption and decryption

Several distinct transformations are applied successively over the data block bits during a set number of iterations, known as rounds, in the encryption and decryption process. There must be ten iterations for a key length of 128 bits. The AES encryption (Cipher) and decryption (Inverse Cipher) procedures for a 128-bit text block are depicted in Figures 3a and 3b. A 4x4 array of bytes known as the "State" serves as the fundamental building block of the AES algorithm. The method is built on the Substitution Permutation network, which implies it consists of a number of connected mathematical processes. These actions consist of: Subbite: Using the S-box table, the Sub Byte transformation is a nonlinear byte substitution that works independently on every byte in the State. S-box table contains 256 numbers (from 0 to 255). Shift Rows: In the ShiftRows transformation, the rows of the State are cyclically left-shifted over different offsets. Specifically, the determination of the offset depends on the order of the row, or in other words, depends on the r-index of that row. Row 0 is not shifted, row 1 is shifted one byte to the left, row 2 is shifted two bytes to the left, and row 3 is shifted three bytes to the left. Mix

Columns: The Mix Columns transformation operates on the State column by column, treating each column as a four-term polynomial. Add Round Key: The Add Round Key phase performs an operation on the State with one between each byte of the State and each byte of the subkey, this procedure is a straightforward XOR. The three high-level steps of the encryption and decryption process are the first round, middle round, and final round, as illustrated in Figures 3a and 3b. There is simply the AddRoundKey procedure in the Initial round stage. Sequence operations SubByte, ShiftRow, MixColumn, and AddRoundKey make up the Middle round step. As a result, after doing 10 rounds the encryption, and decryption process will output the ciphertext, or plain text, respectively. However, the decryption process uses three other inverse operations of SubByte, ShiftRow, and MixColumn are In vSubByte, InvShiftRow, and InvMixColumn. Furthermore, the direction of this process also is inverted with encryption. In the Middle Round step, the decryption performs sequence operations are InvShiftRow, InvSubByte, AddRoundKey, and InvMixColumn. Finally, the Final Round step includes In vShiftRow, InvSubByte, and AddRoundKey operations.

IV. FPGA IMPLEMENTAION

The entire subsystem, which consists of the AES 128IP core and DMA controller interfaces, is named as AES Controller. The architecture of the connecting model between HPS, SDRAM, and RSA Controller is shown in Figure 5. Furthermore, once operational, the interfaces of AES-128 continuously read and write data between AES-128 and SDRAM, so we built two FIFO’s buffer memory to avoid missing data, and placed as shown in Figure 5. Besides, we have build software on Linux operating systems running on ARM core (HPS) to control and manage hardware. Specifically, we use this application to configure the information for the DMA process are the read address, the write address, and the length of data for one transfer. Furthermore, it is used for checking whenever the AES-128 IP work is done and forward this information to the cryptography applications to perform additional processing steps.

V. LITERATURE REVIEW

[1] M. Rajeswara Rao, Dr.R.K.Sharma, SVE Department, NIT Kurushetra “FPGA Implementation of combined S box and Inv S box of AES” 2017 4th International conference on signal processing and integrated networks (SPIN).

An implementation of a combinational memory-less S-Box and inv S-Box (combinely) for ByteSub and InvByteSub transformations of AES on a same hardware. This is a part of the combined architecture of AES in which both encryption and decryption can be performed with an enable pin. Previously LUTs are used to implement the S-Box and Inv S-Box of AES separately, which causes large amount of memory and area. In this paper, the proposed architecture is implementing using composite field arithmetic in finite fields GF (28) which is advantageous than LUT approach on the basis of hardware complexity. As both S-Box and Inv S-Box are implementing on a same hardware, there is large reduction in gate count as well as in area.

[2] Nalini C. Iyer ; Deepa ; P.V. Anandmohan ; D.V. Poornaiah “Mix/InvMixColumn decomposition and resource sharing in AES”.

In this paper, compact architectures for AES Mix Column and its inverse is presented to reduce the area cost in resulting AES implementation. In the hardware implementation of AES with direct mapping substitute byte optimization, MixColumn/Inverse MixColumn transformation demands the utilization of logic resources and then effects the critical path delay and resulting throughput. The proposed MixColumn/Inverse MixColumn design based on byte and bit-level decomposition leads to two types of architecture which demonstrates deeper resource sharing within byte and between bytes and rearrangement of output terms with respect to FPGA architecture in bit level resply. The proposed architectures have been investigated on a FPGA based implementation platform. Application of the proposed architectures resulted in reduction of reconfigurable logic area by 40% as compared to separate implementation of MixColumn and Inverse MixColumn reduction and also path delay by 9% resply. Experimental results show that our proposed architecture can reduce the area cost significantly and compared with other previous implementations reported so far.

Summary: The proposed Mix/Inv mix architecture can reduce the area cost significantly.

[7] Yulin Zhang ; Xinggang Wang; “Pipelined implementation of AES encryption based on FPGA” 2010 IEEE International Conference on Information Theory and Information Security.

This paper presents the outer-round only pipelined architecture for a FPGA implementation of the AES-128 encryption processor. The proposed design uses the Block RAM storing the S-box values and exploits two kinds of Block RAM. By combining the operations in a single round, we can reduce the critical delay. As the network transmission speed upgrades to the gigabits per second (Gbps), the software-based implementations of cryptographic algorithms cannot meet its needs. The hardware-based implementations using some special optimization techniques (such as pipeline and lookup tables, etc.), can greatly improve throughput and reduce the key generation time. Besides, the processes of cryptographic algorithms and the key generation packaged in chip, which cannot easily be read or changed by external attacker, so hardware-based implementations can get the higher physical security .

Summary: By combining the operations in a single round, we can reduce the critical delay.

[10] C. Sivakumar ; A. Velmurugan ; “High Speed VLSI Design CCMP AES Cipher for WLAN (IEEE 802.11i)” 2007 International Conference on Signal Processing, Communications and Networking.

The Advanced Encryption Standard (AES) algorithm has become the default choice for various security services in numerous applications. In this paper, we propose a high speed, non-pipelined FPGA implementation of the AES-CCMP (Counter-mode/CBCMAC Protocol) cipher for wireless LAN using Xilinx development tools and Virtex-It Pro FPGA circuits. IEEE 802.11i defines the AES-based cipher system, which is operated on CCMP Mode. All the modules in this core are described by using Verilog 2001 language. The developed AES CCMP core is aimed at providing high speed with sufficient security. The AES S-box is a 256-entry table composed of two transformations: First each input byte is replaced with its multiplicative inverse in GF (2^8) with the element {00} being mapped onto itself, followed by an affine transformation over GF (2^8). For decryption, inverse S-box is obtained by applying inverse affine transformation followed by multiplicative inversion in G(2^8).

Summary: In the current wireless LAN environment, WEP-the current algorithm for security-is not safe against attacks. We consider AES CCMP algorithms for wireless LAN security.

[14] P. S. Abhijith ; Mallika Srivastava ; Aparna Mishra ; Manish Goswami ; B. R. Singh ; "High performance hardware implementation of AES using minimal resources" 2013 International Conference on Intelligent Systems and Signal Processing (ISSP).

Increasing need of data protection in computer networks led to the development of several cryptographic algorithms hence sending data securely over a transmission link is critically important in many applications. Hardware implementation of cryptographic algorithms are physically secure than software implementations since outside attackers cannot modify them. In order to achieve higher performance in today's heavily loaded communication networks, hardware implementation is a wise choice in terms of better speed and reliability. Encryption is usually done just before sending data. Achieving high throughput for encryption algorithm for a communication channel of high data rate is a challenging task.

Summary: With the designing of all the operations as LUTs and ROMs, the proposed architecture achieves a throughput and thereby utilizing only slices in the targeted FPGA.

[13] N. S. Sai Srinivas ; Md. Akramuddin; "FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption" 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).

AES algorithm or Rijndael algorithm is a network security algorithm which is most commonly used in all types of wired and wireless digital communication networks for secure transmission of data between two end users, especially over a public network. This paper presents the hardware implementation of AES Rijndael Encryption and Decryption Algorithm by using Xilinx Virtex-7 FPGA. The hardware design approach is entirely based on pre-calculated look-up tables (LUTs) which results in less complex architecture, thereby providing high throughput and low latency. There are basically three different formats in AES. They are AES-128, AES-192 and AES-256. The encryption and decryption blocks of all the three formats are efficiently designed by using Verilog-HDL and are synthesized on Virtex-7 XC7VX690T chip (Target Device) with the help of Xilinx ISE Design Suite-14.7 Tool. The synthesis tool was set to optimize speed, area and power. The power analysis is made by using Xilinx XPower Analyzer. Pre-calculated LUTs are used for the implementation of algorithmic functions, namely S-Box and Inverse S-Box transformations and also for GF (28) i.e. Galois Field Multiplications involved in Mix-Columns and Inverse Mix-Columns transformations. The proposed architecture is found to be having good efficiency in terms of latency, throughput, speed/delay, area and power.

Summary: The LUT based design approach gives less complex architecture and saves the processing time to a great extent by retrieving the necessary values from memory locations.

[16] Ashwini M. Deshpande ; Mangesh S. Deshpande ; Devendra N. Kayatanavar; "FPGA implementation of AES encryption and decryption" 2009 International Conference on Control, Automation, Communication and Energy Conservation

Advanced Encryption Standard (AES), a Federal Information Processing Standard (FIPS), is an approved cryptographic algorithm that can be used to protect electronic data. The AES can be programmed in software or built with pure hardware. However Field Programmable Gate Arrays (FPGAs) offer a quicker and more customizable solution. This paper presents the AES algorithm with regard to FPGA and the Very High Speed Integrated Circuit Hardware Description language (VHDL). ModelSim SE PLUS 5.7g software is used for simulation and optimization of the synthesizable VHDL code. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 8.2i suite. All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption. Xilinx XC3S400 device of Spartan Family is used for hardware evaluation. This paper proposes a method to integrate the AES encrypter and the AES decrypter. This method can make it a very low-complexity architecture, especially in saving the hardware resource in implementing the AES (Inv) Sub Bytes module and (Inv) Mix columns module etc. Most designed modules can be used for both AES encryption and decryption. Besides, the architecture can still deliver a high data rate in both encryption/decryption operations. The proposed architecture is suited for hardware-critical applications, such as smart card, PDA, and mobile phone, etc.

Finally, the normalized Hamming distance algorithm is used for matching retrieval. In order to protect the speech security in the cloud, a speech encryption algorithm based on a 4D hyper chaotic system is proposed. The experimental results show that the proposed method has good discrimination, robustness, recall and precision compared with the existing methods, and it has good retrieval efficiency and retrieval accuracy for longer speech.

Summary: This method can make it a very low-complexity architecture and low hardware utilization.

VI. EXPERIMENTAL RESULTS

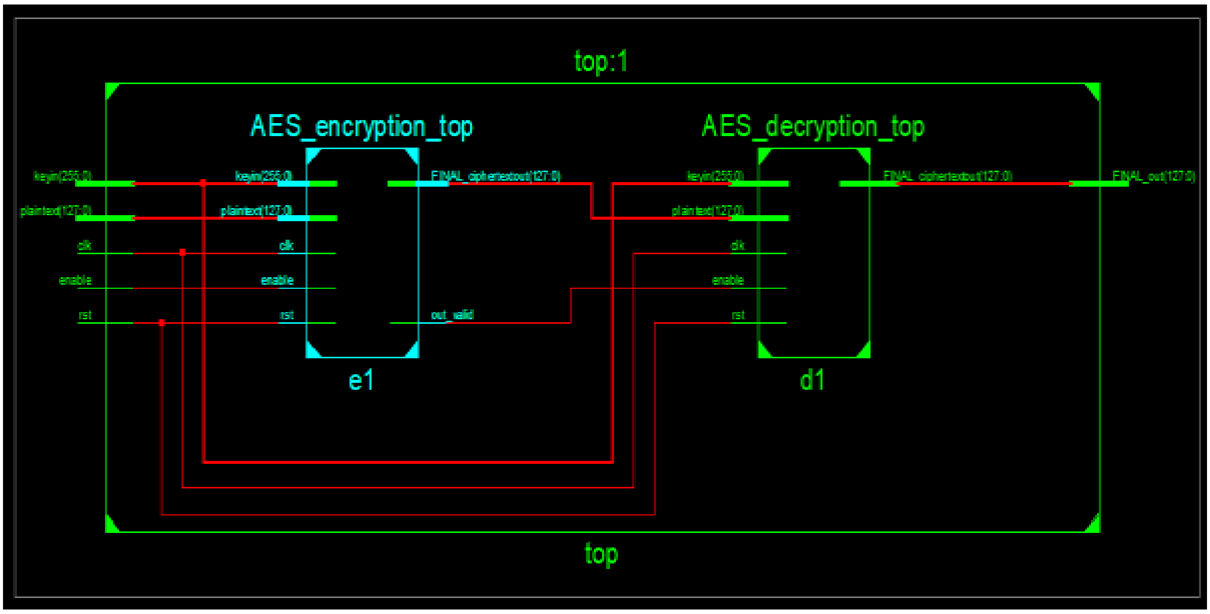
A. Simulation results

Figure 6 presents the simulation results of AES-128 with the ModelSim software. Specifically, shows the simulation result of the encryption, and shows the outcome of decryption. Besides, the red rectangles are used for marking the input of the secret key, which is fetched parallelly. Additionally, the blue and yellow rectangles marked the input data and output data, which are loaded and wrote serially, respectively. The data used for simulation are:

Secret key: 61626364656667686162636465666768HEX.

Plaintext: 507269736f6e2067616e677320617265HEX.

Ciphertext: e7e4e1139e95ebe58caf23960a813f41HEX.



In the encryption process, Figure 6, the input data is plaintext, and output data is ciphertext. While in the decryption process, the input data is ciphertext, and the output is plaintext. Through the simulation processes, we can verify that the AES IP core can convert ciphertext to plaintext by the encryption, and it can convert ciphertext back to plaintext by the decryption process precisely. Through the simulation process, we have demonstrated the function of encryption and decryption of our IP is correct.

VII. CONCLUSION

In this study, we have developed a high-performance, small-sized cryptosystem that can manage the cryptographic operations of Internet of Things systems, based on SoC FPGA platforms. Two primary components of our system are FPGA and HPS. We fully self-implemented an IP core on the FPGA side to carry out AES-128 symmetric cryptography. To operate and administer the hardware system, we also developed special Linux kernel and drivers on the HPS side. Through our work, we have assessed a few implementation cryptography algorithms and approaches on FPGA for high-speed processing, including DMA, Pipeline, and custom memory modules. Additionally, our system can react to the dependability of the security of the Internet of Things without worrying about malware on both a hardware- and software-based level thanks to self-implementation. Furthermore, our cryptosystem—which is built on SoC FPGA platforms—has several benefits, including low cost, low power consumption, and high performance, making it appropriate for use as node controllers or gateway devices in Internet of Things systems.

References

[1] M. Rajeswara Rao, Dr.R.K.Sharma, SVE Department, NIT Kurushetra “FPGA Implementation of combined S box and Inv S box of AES” 2017 4th International conference on signal processing and integrated networks (SPIN).

[2] Nalini C. Iyer ;Deepa ; P.V. Anandmohan ; D.V. Poornaiah “Mix/InvMixColumn decomposition and resource sharing in AES”.

[3] Xinmiao Zhang, Student Member, IEEE, and Keshab K. Parhi, Fellow, “High Speed VLSI architectures for the AES Algorithm”, IEEE. VOL.12. No.9. September 2004

[4] ShrivathsaBhargav, larry Chen, abhinandanMajumdar, Shiva Ramudit “128 bit AES Decryption”, CSEE 4840 – Embedded system Design spring 2008, Columbia University.

[5] Atul M. Borkar ; R. V. Kshirsagar ; M. V. Vyawahare “FPGA implementation of AES algorithm”.

[6] Announcing the ADVANCED ENCRYPTION STANDARD (AES), November 26 2001.

[7] Yulin Zhang ; Xinggang Wang; “Pipelined implementation of AES encryption based on FPGA” 2010 IEEE International Conference on Information Theory and Information Security.

[8] Yuwen Zhu ; Hongqi Zhang ; YibaoBao ; “Study of the AES Realization Method on the Reconfigurable Hardware” 2013 International Conference on Computer Sciences and Applications.

[9] Tsung-Fu Lin ; Chih-Pin Su ; Chih-Tsun Huang ; Cheng-Wen Wu; “A high-throughput low-cost AES cipher chip” Proceedings. IEEE AsiaPacific Conference on ASIC.

[10] C. Sivakumar ; A. Velmurugan ; “High Speed VLSI Design CCMP AES Cipher for WLAN (IEEE 802.11i)” 2007 International Conference on Signal Processing, Communications and Networking.

[11] VatcharaSaicheur ;KrerkPiromsopa ; “An implementation of AES128 and AES-512 on Apple mobile processor” 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)

[12] S.P Guruprasad ; B.S Chandrasekar ; “An evaluation framework for security algorithms performance realization on FPGA” 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)

[13] N. S. Sai Srinivas ; Md. Akramuddin; “FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption” 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).

[14] P. S. Abhijith ; Mallika Srivastava ; Aparna Mishra ; Manish Goswami ; B. R. Singh ; “High performance hardware implementation of AES using minimal resources” 2013 International Conference on Intelligent Systems and Signal Processing (ISSP).

[15] Wei Wang ;Jie Chen ; Fei Xu ; “An implementation of AES algorithm Based on FPGA” 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery

[16] Ashwini M. Deshpande ;Mangesh S. Deshpande ; Devendra N. Kayatanavar; “FPGA implementation of AES encryption and decryption” 2009 International Conference on Control, Automation, Communication and Energy Conservation.