IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CRYPTOGRAPHIC HARDWARE FOR EMBEDDED SYSTEMS

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

[1]Professor, Department of Electronics and Communication, Cambridge Institute of Technology (CITech), KR Puram, Bengaluru, India.

[2,3,4,5]Students, Electronics and Communication Engineering, Cambridge Institute of Technology, Bengaluru, India

Abstract: Developing a cryptography model that can translate legible writing from common language and converse is the aim of this research. For SoC level FPGA, the AES 128-bit symmetric cryptography technique is used to avoid malware in both hardware and software. The ideal AES design with an enhanced security scheme is also described in this study. The suggested model enables pipelined reusing of the same hardware. Furthermore, suggests expanding the use of dynamic key extraction in cryptosystems to increase randomness through the addition of outer layer security. A completely automated process for creating keys framework utilizing digital biometrics is utilized to produce a 256-bit key with enhanced randomness for 32 From the provided digital biometric, pixels are randomly selected, and the values of all the selected pixels are concatenated at random to generate a 256- bit encryption key.

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).*

I. INTRODUCTION

The practice of turning regular, daily English writing into incomprehensible text and vice versa is known as cryptography. Hash functions, public key cryptography, and symmetric key cryptography are the three categories of cryptographic techniques. It requires less computing power and may be implemented faster and easier. An explanation of the symmetric key block known as the AES in December 2001. 128 bits is encrypted and decrypted using this non-Feistel block cipher. Three distinct key lengths are offered. Ten processing rounds for 128-bit keys, twelve rounds for 192-bit keys, and fourteen rounds for 256-bit keys make up the encryption/decryption process. Embedded cryptography devices based on FPGA are simply encryption and decryption models. We constructed a 256-bit key-based AES for 128-bit data. It is significantly optimized in terms of area, power, significant timeframe, and sub-byte blocks. An expanded model provides a generic architecture and a feasible way to apply the AES algorithm with a biometric key. The recommended method outperformed using static keys in terms of security. The functional verification of the existing AES algorithm and the suggested AES algorithm is carried out with Xilinx 14.7 version. Modern block ciphers employ substitution-boxes to convert plaintext data nonlinearly and generate suitably perplexing cipher- text data. It is commonly recognized that the underlying substitution-boxes play a major role in determining the cryptographic strength and security of these block ciphers. for the obvious reason that they are the only ones in charge of adding the necessary nonlinearity and complexity to the security system, which could irritate opponents. Therefore, a number of concepts have been investigated in order to build robust S-boxes. This

work investigates a novel basic modular technique for the very first, aiming at creating a nonlinear S-box with the same objective. The new modular approach is composed of three operations: new transformation, modular inverses, and permutation. By varying the specific transformation parameters, producing several extremely nonlinear S-boxes can be achieved with ease. By comparing it to recent S-boxes and assessing its critical performance against a set of benchmarking criteria, such as high nonlinearity, lack of fixed points, fulfillment of SAC and BIC properties, low differential uniformity, and linear approximation probability, we present an example S-box whose upright cryptographic potential is demonstrated. Additionally, an image encryption method that provides high statistical and differential encryption performance is demonstrated by using the recently constructed S-box to execute pixel substitution and shuffling.

II. LITERATUREREVIEW

Table1.LiteratureReviewof Cryptographic Hardware for Embedded Systems with various existing solutions

ıblished year	ournal Type	Author Name	Title of the paper	Outcomes
2017 [1]	IEEE	M. Rao, Rajeswar a Dr.R.K.Sharma, SV E Department, NIT Kurushetra	FPGA Implementation of combined S box and Inv S box of AES	By comparing with the LUT based in this hardware complexity is reduced.
2012 [2]	IEEE	Nalini C. Iyer ; Deepa ; P.V. Anandmohan ; D.V. Poornaiah	Mix/Inv Mix Column decomposition resource , sharing in AES	The proposed Mix/Inv mix architecture can reduce the area cost significantly.
2010 [3]	IEEE	Yulin Zhang ; Xinggang Wang;	Pipelined implementation of AES encryption based on FPGA	By combining the operations in a single round, we can reduce the critical delay
2020 [4]	IEEE	Tsung-Fu Lin ; Chih-Pin Su ; Chih-Tsun Huang ; Cheng- Wen Wu	A High-Throughput Low- Cost AES Ciphera Chip	Our pipelined design has very high throughput rate
2013 [5]	IEEE	P. S. Abhijith ; Mallika Srivastava ; Aparna Mishra ; Manish Goswami ; B. R. Singh	High Performance Hardware Implementation of AES Using Minimal Resources	Sub Byte, Inverse Sub Byte, Mix Column, and Inverse Mix operations are used to gain more performance and less area.
2020 [6]	IEEE	Shahbazi, Karim, and Seok- Bum Ko	Area-efficient nano-AES implementation for Internet- of-Things devices	Execution time is reduced and to reduce power consumption clock gating technique is used
2021 [7]	IEEE	Yang, Cheng-Hsiung, and Yu-Sheng Chien;	Implementation and Design of a Hybrid Chaos-AES Color Image Encryption	Here signal transmission of all nodes in the combinational part and the topological structure error

			Algorithm	propagation .
2018 [8]	IEEE	Wang, H., Forsmark, S., Brisfors, M., & Dubrova, E.	Multi-Source Training Deep-Learning Side-Channel Attacks	In this papers they discussed about bit-stream extraction for IP core theft & reverse engineering process.
2004 [9]	IEEE	Xinmiao Zhang	High Speed VLSI architectures for the AES Algorithm	These architectures are tailored to execute AES encryption and decryption operations swiftly and effectively catering to the , demands of modern cryptographic applications.

III. METHODOLOGY

Proposed System: The proposed approach uses both encryption and decryption for same S-box hardware. The one element that sets apart is inverse S-box (decryption) and S-box (encryption) is the Affine transform. For both the S-box and the inverse S-box, all extra encryption and decryption circuitry is identical. Therefore, we apply the identical logic to both encryption and decryption, with the exception of the Affine transform. choosing a mux between the inverse S-box and the S-box. S-box paths for AES encryption and decryption are selected using the inverse S-box path, respectively. The way "Add round key" and "Mix column" are integrated in the recommended design is referred to as "Mix block".

Figure 1: System architecture

System design:

The suggested implementation of 32-bit AES. In every cycle, we perform operations on a word (32 bits). The following count of blocks are needed for both 32-bit and 128-bit implementations.

1) S box: 16 pieces, 4 clock cycles.
2) Four mix blocks of columns, one forever.

Structure combining an inverse S box and a S box:

One aspect of the suggested 32-bit operating paradigm is the reuse of the S-box and Mix Column blocks. "Mix block" refers to the way that "Add round key" and "Mix column" are integrated in the proposed design.

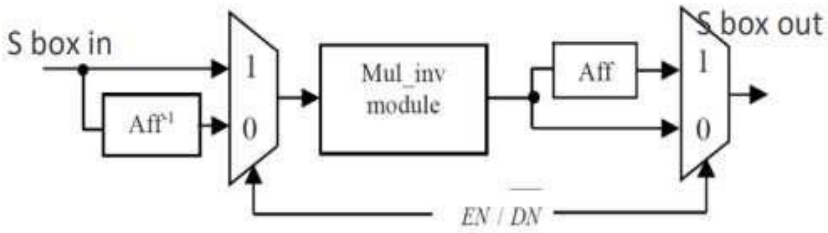


Figure 2: Structure combining an inverse S box and a S box

Combined Mix Column Structure with Round Key – Mix

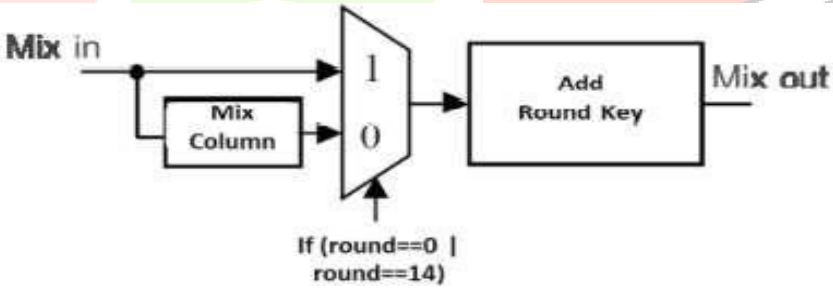


Figure 3: Combined Mix Column Structure with Round Key – Mix

Pipelined structure:

The pipeline structure, When each hue denotes a distinct round in the following manner:

S box	Shift	Mix	Cycle
		Mix_0	1
Sub_0	-	Mix_1	2
Sub_1	Shift_0	Mix_2	3
Sub_2	Shift_1	Mix_3	4
Sub_3	Shift_2	-	5
Key_?	Shift_3	Mix_0	6
Sub_0	-	Mix_1	7
Sub_1	Shift_0	Mix_2	8
Sub_2	Shift_1	Mix_3	9
Sub_3	Shift_2	-	10
Key_3	Shift_3	Mix_0	11
Sub_0	-	Mix_1	12
Sub_1	Shift_0	Mix_2	13
Sub_2	Shift_1	Mix_3	14
Sub_3	Shift_2	-	15
-	Shift_3	Mix_0	16
.	.	Mix_1	17
Key_14	.	Mix_2	18
Sub_0	-	Mix_3	19
Sub_1	Shift_0	.	.
Sub_2	Shift_1	.	.
Sub_3	Shift_2	-	.
-	Shift_3	Mix_0	71
		Mix_1	72
		Mix_2	73
		Mix_3	74

Table 2:

Round 0: Mix;
Round 1: Mix;
Round 2: Mix; Round 3: Mix; and Round 14: Mix

There are 32 bits in each word. We are performing the Mix working on word 0 (mix_0) in cycle 1. The symbol for this would be cycle 1 [round 0 (mix 0)].
This 32-bit word can therefore be used to carry out a 32-bit Sub operation. Consequently, word 0's sub operation (sub_0) and word 1's mix operation are being performed in cycle 2. The four words in the "mix" block operation do not need to wait to finish because we have valid input for "sub" block word 0 in clock cycle2. Cycle2 [round0(mix_1), round1(sub_0)] would be the representation for this.

In the third clock cycle, We are doing three operations: the mix operation for word 2, the shift operation for word 0 (shift_0), and the sub operation for word 1 (sub_1). For simplicity, we can call this Cycle 3[round1 (sub_1, shift_0, round0(mix_2)]].

Clock cycle 4 consists of sub operations (sub_2), shift operations (shift_1), and mix operations (mix_3) on words 2 and 3. You can describe this as cycle 4 [round0(mix_3), round1 (sub_2, shift_1)]. as well as mix operations in clock cycle 4 on word 3 (mix_3). The representation of this would be cycle 4 [round 0 (mix 3), round 1 (sub_2, shift 1)]. Cycle 5 does not involve a mix operation because the 128-bit (4-word) round 0 mix method is complete. Shift operations for word 2 (shift_2) and sub operations for word 3 (sub_3) are being performed in clock cycle 5. Cycle 5[round1(sub_3, shift_2)] is this, assuming that makes sense.

We won't have to wait for another shift cycle to begin mixing as a consequence. This is cycle 6[round1(shift_3, mix_0)], if we may so name it.. We are doing the mix operation for word 1 and the sub operation (sub_0) for

word 0 in clock cycle 7. Cycle 7[round1(mix_1), round2(sub_0)] would be the depiction of this..
Clock cycle 8 involves sub operations (shift_0 and sub_1) for words 1 and 0 and mix operations (mix_2) for words 2 and 0. This can be written as Cycle8[round2(sub_1, shift_0), round1(mix_2)]

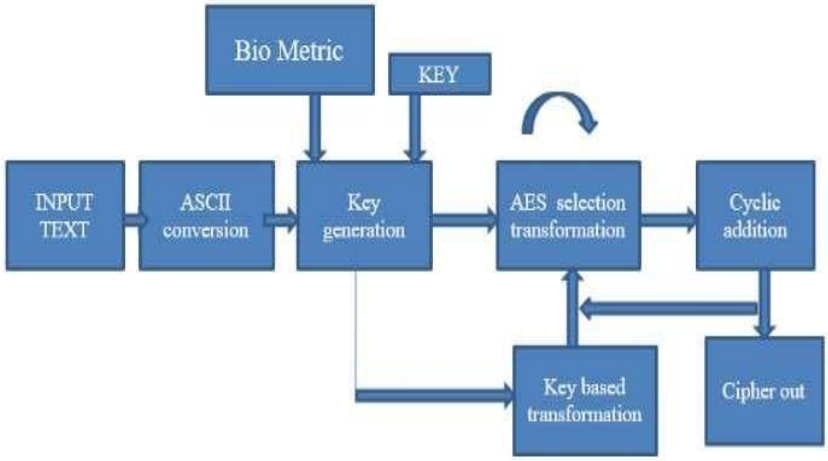
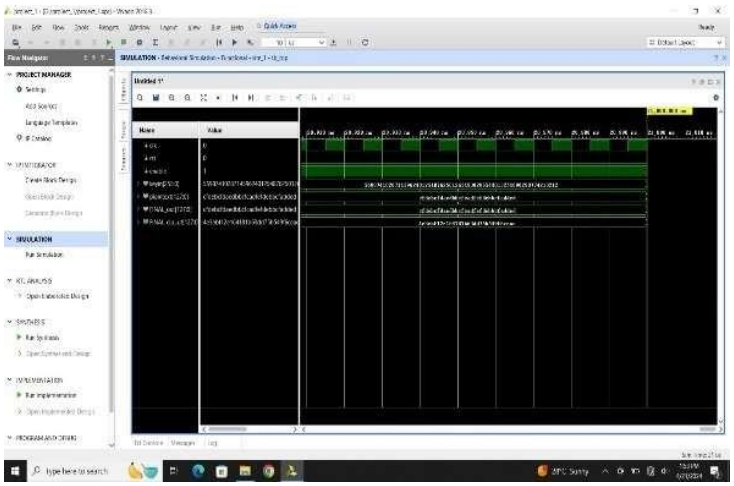


Figure 4: Block Diagram

IV. RESULTS AND DISCUSSION

Cryptographic hardware operates as a sophisticated mechanism for both encrypting and decrypting data, enhancing its safety and security. In this model, plaintext undergoes encryption to transform it into ciphertext, effectively safeguarding sensitive information. In our implementation, a 128-bit input data is subjected to encryption using a 256-bit key, resulting in a ciphertext output, as illustrated in Figure 5. Employing symmetric AES encryption, we attain the synthesized output depicted in Figure 6.

When dealing with image encryption, a grayscale image is specifically chosen, and its pixels are converted into a hexadecimal file format using MATLAB. This hexadecimal representation serves as the input text file for the ISE Design software. Through this process, the image is encrypted, yielding the encrypted image output showcased in Figure 7, along with the synthesis outcome of the top-level module, presented in Figure 8.



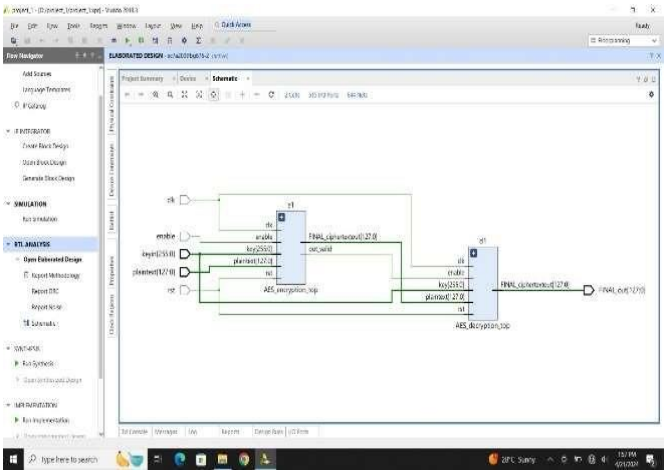
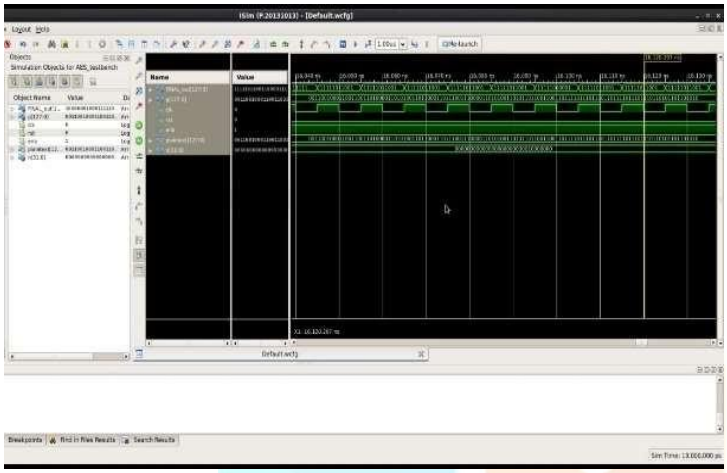


Figure 5: Bit Encryption

Figure 6: Synthesis



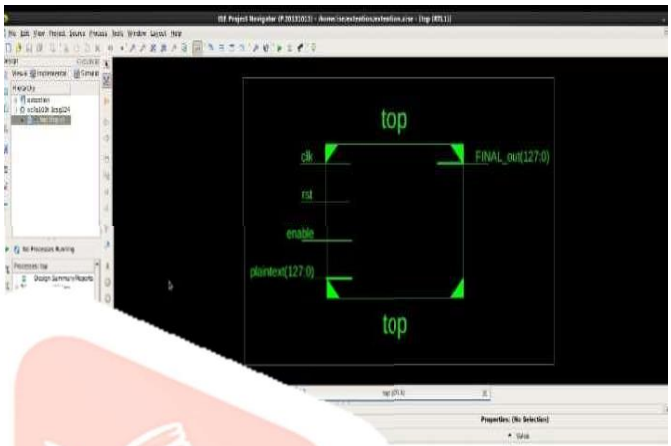


Figure 7: Image Encryption

Figure 8: Synthesis

Conclusion

In this study, throughout this investigation, we have created a small, high-performance cryptosystem that can supervise the cryptographic operations of Internet of Things systems, based on SoC FPGA platforms. Our system consists of two main parts: FPGA and HPS. On the FPGA side, we completely self-implemented an IP core to perform AES-128 symmetric cryptography. We also created unique Linux kernels and drivers on the HPS side to be able to manage and operate the hardware system. We have evaluated a few implementation cryptography techniques and algorithms, including as DMA, Pipeline, and custom memory modules, on FPGA for high-speed processing through our work. Furthermore, because of its self-implementation, our system is able to respond to the Internet of Things' reliability in terms of security without having to worry about malware on the hardware or software front. Additionally, our cryptosystem, which is based on SoC FPGA platforms, is suitable for usage as node controllers or gateway devices in Internet of Things systems because of its inexpensive price, low power consumption, and excellent performance.

REFERENCE:

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

[4] Tsung-Fu Lin ; Chih-Pin Su ; Chih-Tsun Huang ; Cheng-Wen Wu; "A High-Throughput Low-Cost AES Cipher Chip" 2013 International Conference on Computer Sciences and Applications.

[5] 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).

[6] Shahbazi, Karim, and Seok-Bum Ko; "Area-efficient nano-AES implementation for Internet-of-Things devices"

[7] Yang, Cheng-Hsiung, and Yu-Sheng Chien; "Implementation and Design of a Hybrid Chaos-AES Color Image Encryption Algorithm".

[8] Wang, H., Forsmark, S., Brisfors, M., & Dubrova, E; "Multi-Source Training Deep-Learning Side-Channel Attacks" Yuwen Zhu ; Hongqi Zhang ; YibaoBao ; "Study of the AES Realization Method on the Reconfigurable Hardware" 2013 International Conference on Computer Sciences and Applications.

[9] 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