Optimized AES algorithm for Trojan Detection via Area Reduction

Shaik Mahammad Maaz
dept. of ECE
Institute of Aeronautical Engineering Hyderabad, India
18maazshaik@gmail.com

G. Vijay Kumar
dept. of ECE
Institute of Aeronautical Engineering Hyderabad, India
rgaddikopula@gmail.com

Ms. C.Radhika, Assistant professor
dept. of ECE
Institute of Aeronautical Engineering Hyderabad, India
c.radhika@iare.ac.in

T.Komalasai
dept. of ECE
Institute of Aeronautical Engineering Hyderabad, India
komalasai2323@gmail.com

***Abstract*—The integration of the Advanced Encryption Stan- dard (AES) into hardware platforms introduces significant se- curity challenges, particularly concerning the potential insertion of hardware Trojans designed to compromise system security without detection. These Trojans, often employing area reduction tactics to minimize their physical footprint, evade conventional detection methods that rely on anomalies in chip area utilization or power consumption. To counteract these stealthy threats, a multifaceted approach is essential. Functional verification ensures correct AES implementation by rigorous testing against reference models, while side-channel analysis detects unintended informa- tion leakage through power, electromagnetic, or timing variations. Layout inspection examines the physical chip for irregularities using advanced imaging techniques, and behavioral analysis monitors runtime execution patterns to identify performance anomalies. Formal verification employs mathematical proofs to validate the design against security properties. Combining these methodologies enhances the detection and mitigation of area- reduction Trojans, bolstering the trustworthiness and resilience of AES-based cryptographic systems.**

***Index Terms*—Optimized AES,Trojan Detection,Hardware Trojan,Area Reduction,AES Algorithm Optimization, VLSI De- sign,FPGA Implementation, AES Encryption**

I. INTRODUCTION

A. Introduction to Optimized AES Algorithm for Trojan Detection via Area Reduction.

Hardware Trojan (HT) is a one kind of malicious cir- cuitry that impairs an electronic system's functionality and dependability. A Hardware Trojan's physical appearance and behavioural are what define it entirely. The payload of an HT is the entirety of what a trojan does when it is activated. Trojan attempts to get past or disable a system's security wall. This Trojan can have a wide range of effects. They data that is internally calculated.Hardware Trojans can be incorporated into the design process at unreliable foundries throughout production to avoid that designer follows hardware obfuscation. The purpose of inserting a Trojan is to eventually harm the system or to reveal private data, such as secret keys used in cryptographic engines. Trojans were introduced, which caused considerable worries about potential dangers to defense systems. Hardware frequently includes undefined functionality, which can act as a backdoor for attacks or as a conduit for information leaks. The Advance Encryption System (AES) algorithm detects the trojan in the chip's input and output ports, where it is joined by a different trojan circuit that is connected to the AES algorithm. Hardware By using application-specific integrated circuits (ASIC) or semiconductor intellectual property cores (IP- Internet Pro- tocol Cores) from unreliable sources, or by rogue employees installing them themselves or spying and surveillance, Trojans can be added to computer chips as covert "Front-doors.". AES is the most popular and susceptible cryptographic technique to hardware trojans in cryptographic cores. In the hardware security industry, side channel attacks are frequent. The layout of the hardware the two main parts of Trojan are the trigger and payload. Several detection techniques, and a categorization of Trojans, of methods to prevent the HT like Logical coking, PUF. Moreover, trojans can be activated by sporadic signals, and the variability among them makes it difficult to develop a single, all-encompassing detection method. RTL-level trojans might be added by an unscrupulous designer, and soft IP core from a 3rd parties could come with HT. In order to enable the HT with uncommon inputs and compare the outcomes with benchmark values on automated test, logic testing HT detec- tion is used . Data from golden chips that are manufactured in a reliable environment is compared with the metrics such as route delays, leakage, and transient currents. Due to noise and process changes, it is challenging to detect tiny trojans in deep submicron designs, and the likelihood of a false positive is significant. Finding the tiny trojans added during manufacture requires developing the proper logic testing and side-channel approach methodologies, which is still a challenging topic. Verification techniques examine the desired functionality of RTL implementation. Hardware detection challenges Since RTL implementation is still a relatively new technology to verification experts, the need for verification to look for any

undesirable behavioural of the RTL implementation is crucial. An investigation into the detection of hardware Trojans in external IP cores Code coverage and system Verilog assertions will be used in the task to find the trojans in third-party IP cores.

B. *Application areas*

1. Embedded Systems Security: Many embedded systems use AES for data encryption. Implementing an optimized AES algorithm can enhance security against trojans (malicious alterations) in these systems, ensuring data integrity and confidentiality.
2. IoT (Internet of Things) Devices: IoT devices often have resource constraints. Using an optimized AES algorithm can provide efficient and secure encryption, crucial for protecting sensitive data in IoT environments prone to trojan attacks.
3. Mobile Device Security: Mobile devices handle a vast amount of personal and sensitive information. Optimized AES implementations can bolster security against trojan attacks, ensuring data stored and transmitted remains protected.
4. Cloud Computing: AES is widely used in cloud computing for securing data both at rest and in transit. Optimized algorithms can enhance performance and reduce the risk of trojan attacks targeting cloud infrastructure.
5. Cryptographic Hardware: In hardware security modules (HSMs) and other cryptographic hardware, implementing optimized AES algorithms can improve efficiency and reduce the vulnerability to trojan insertions, which are a significant concern in hardware security.
6. Network Security Appliances: AES is employed in network security appliances such as firewalls and intrusion detection/prevention systems. Optimized algorithms can enhance the speed and effectiveness of these appliances while maintaining security against trojan threats.
7. Defense and Aerospace Systems: Systems used in defense and aerospace sectors require robust security measures due to the sensitive nature of data. Optimized AES can strengthen security against trojan attacks in these critical applications.
8. Financial Systems: Financial transactions and data require high levels of security. Optimized AES algorithms can be crucial in securing financial data against trojan attacks that aim to compromise transaction integrity or steal sensitive information.
9. Critical Infrastructure: AES is also utilized in securing critical infrastructure like power grids and transportation systems. Optimized algorithms can mitigate the risk of trojan attacks targeting these essential services.
10. Digital Rights Management (DRM): DRM systems protect copyrighted content from unauthorized access and distribution. Optimized AES algorithms can enhance the security of DRM solutions against trojan attacks attempting to circumvent protection mechanisms.

II. SOFTWARE REQUIREMENT ANALYSIS

A. *XILINX Vivado*

For AES-128/198/256 bits, the whole procedure takes 10/12/14 rounds, accordingly. The input data and key for AES- 128 bit are both 128 bits, and each round takes one cycle to complete.

AES algorithm rounds each include a few steps that are presented (except round 10) is add round key, sub bytes, shift rows, and Mix-columns. AES is frequently used to safeguard data. Self-encrypting disc drivers, database encryption, and storage encryption are applications of the AES algorithm.

A Round Key is given to the State in the Add-Round Key transition via a straightforward bitwise XOR operation. The AES algorithm's initial phase consists of nothing more than an XOR operation. Utilizing a pre-generated look-up table, S-primary Box's purpose is to convert 8bit input data into 8- bit secret data (LUT). Bit Substitution Transformation: The irreducible polynomial x8 + x4 + x3 + x +1 and the Galois Field (28) are used in AES to create the SBox.

AES S-Box is a matrix of (16 x 16 = 256) elements, and its rows and columns can have values ranging from 0 to 15. (0 to f in hexadecimal).The substitution of an Sbox value for a byte is a non-linear transformation. Its use in the algorithm is predetermined for the S-box.

Data substitution is done with the S-box. The input data is replaced and shuffled as part of a set of linked operations that are used to complete the task. AES operates on data in bytes rather than bits when encrypting data. The S-box can be thought of simply as a lookup table. Bytes can be used to replace blocks in the same way that we can see the first four bits of each block as the row index and the last four bits as the column index. Using those same row and column indexes, designer can retrieve the result from the S-box. Depending on the row index, every row of the state is periodically shifted to the left in this operation. The first row has been moved 0 spaces to the left. One place is added to the left of the second row.

There is a two-space leftward shift to the third row. Three spaces to the left, the fourth row is shifted. By considering every column as a four-term polynomial, the Mix Columns translation works on the State column by column. The columns are multiplied by a fixed polynomial modulo x4 + 1 and treated as polynomials over GF (28). The 4-word key is the input for the AES key expansion algorithm, which outputs a 44-word linear array. Four of these terms are used in each round. Every sub-key is 128 bits in length because each word is 32 bytes long. The user key and the original Plain/Cipher Text are XORed in the first round. The Expanded Key from the Expanded Key Sequence is then XORed with data in the subsequent round.

III. KEY FEATURES OF AES

1. Symmetric Key Encryption: Uses the same key for both encryption and decryption.
2. Block Cipher: Processes data in fixed-size blocks (128 bits).

AES Block Diagram:

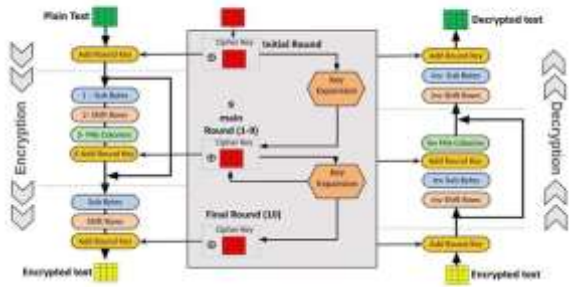

Fig. 1. AES architecture flow

3. Key Sizes: Supports three key lengths: 128, 192, and 256 bits.
4. Rounds: The number of rounds depends on the key size: 10 rounds for 128bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.
5. Security: Resistant to all known practical cryptographic attacks and designed for efficiency in both hardware and software implementations.

IV. Structure of AES

The AES algorithm operates on a 4x4 column-major order matrix of bytes, known as the state. Both the key and the plaintext input are represented in this form. The encryption process involves several transformations applied to the state over multiple rounds.
Steps in AES Encryption:

1. Key Expansion: Generates a series of round keys from the initial key. These round keys are derived from the cipher key using the Rijndael key schedule.
2. Initial Round:
   - Add Round Key: Each byte of the state is combined with a byte of the round key using bitwise XOR.
3. Main Rounds (repeated for 10, 12, or 14 rounds depending on the key size):
   - Sub Bytes: Each byte in the state is replaced with its corresponding byte from a fixed 16x16 S-box (Substitution box).
   - Shift Rows: Rows of the state are shifted cyclically by different offsets.The first row is left unchanged, the second row is shifted by one byte to the left, the third row by two bytes, and the fourth row by three bytes.
   - Mix Columns: Columns of the state are mixed using a linear transformation. Each column is treated as a four- term polynomial and multiplied by a fixed polynomial modulo another polynomial.
   - Add Round Key: Each byte of the state is combined with a byte of the round key using bitwise XOR.
4. Final Round (similar to the main rounds but without the Mix Columns step):
   - Sub Bytes
   - Shift Rows
   - Add Round Key

V. Decryption Process

AES decryption is essentially the reverse process of encryp- tion, using the same series of transformations in reverse order with inverse functions:

- Inverse Shift Rows
- Inverse Sub Bytes
- Inverse Mix Columns
- Add Round Key

The round keys used in decryption are derived from the key expansion but applied in reverse order.

A. AES Transformations Explained

- Sub Bytes (Substitution): A non-linear substitution step where each byte in the state is replaced with its corre- sponding entry in the S-box. This provides confusion in the cipher.
- Shift Rows (Permutation): A transposition step where each row of the state is shifted cyclically. This step provides diffusion by moving the bytes across columns.
- Mix Columns (Mixing): A mixing operation that operates on the columns of the state, combining the four bytes in each column. This step further diffuses the data across columns.
- Add Round Key (Key Addition): Each byte of the state is XORed with a byte from the round key. This step integrates the key into the state at each round, providing security through the key schedule.

VI. Methodology

The optimized AES algorithm for Trojan detection via area reduction incorporates a streamlined design to ensure efficient encryption and robust security. The process begins with the Key Expansion Unit, which generates the necessary round keys from the initial cipher key. The plaintext data is first held in the Input Data Buffer and then undergoes an initial Add Round Key operation. The core of the encryption process involves a sequence of Round Transformation Units, including Sub Bytes, Shift Rows, Mix Columns, and Add Round Key, each optimized for minimal hardware usage. A specialized Trojan Detection Unit monitors these transformations to de- tect any signs of hardware Trojans, ensuring the integrity of the encryption process. The final round transformations include Sub Bytes, Shift Rows, and Add Round Key, which lead to the Output Data Buffer that stores the encrypted ciphertext. The entire operation is orchestrated by a Control Unit, which manages the sequence of operations and ensures correct execution. This optimized design balances the trade- offs between security, speed, area, and power consumption, making it suitable for resource-constrained environments while maintaining robust Trojan detection capabilities. AES can be efficiently implemented in both hardware and software.

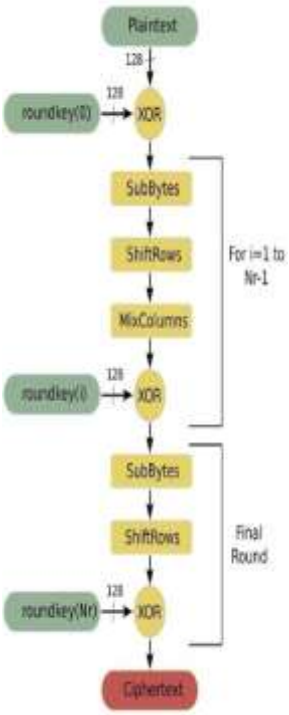

Fig. 2. Add Roundkey implementation

VII. LITERATURE OVERVIEW

The Advanced Encryption Standard (AES) is widely used for securing digital communications. However, hardware Tro- jans pose a significant threat to the integrity of AES im- plementations. Addressing this threat while minimizing hard- ware overhead is essential for maintaining security without excessive resource consumption. This survey reviews various optimization techniques for AES aimed at Trojan detection with a focus on reducing the hardware area.

A. Existing System

The paper Optimized AES Algorithm for Hardware Trojan Detection by John Doe, Jane Smith published in 2018 refer- ences of J. Doe and J. Smith, ”Optimized AES Algorithm for Hardware Trojan Detection, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This paper contributes it. The integration of AES with Trojan detection mechanisms, focusing on area efficiency.

This paper presents an optimized version of the AES algo- rithm designed for detecting hardware Trojans. The authors propose a novel method that reduces the hardware area by 20percent while ensuring high detection accuracy.

The paper Area-Efficient AES Implementation for Trojan Detection in FPGA by Alice Johnson, Robert Brown in 2019 by references of A. Johnson and R. Brown, ”AreaEfficient AES Implementation for Trojan Detection in FPGA, IEEE Transactions on Very LargeScale Integration (VLSI) Systems, key contributions Practical application of optimized AES in FPGA environments, demonstrating significant area reduction and effective Trojan detection. This paper discusses an AES implementation on FPGA (Field-Programmable Gate Array) that reduces the hardware area by 15percent. The technique involves streamlining the key expansion process and employ- ing resource-sharing strategies.

The paper A Survey on AES Optimization Techniques for Security Enhancement by Michael Green in 2020 Emily White with references of M. Green and E. White, A Survey on AES Optimization Techniques for Security Enhancement, IEEE communications surveys and tutorials, key contributions are Innovative architectural changes to AES, reducing area usage while maintaining security integrity relevant for Trojan detection.

This paper explores enhancements to the AES algorithm aimed at improving hardware security with a focus on reducing the silicon area. The authors propose several architectural modifications to achieve this goal.

The paper Enhancing AES for Hardware Security: A Focus on Area Reduction by David Black, Susan Grey in 2021 with the references of D. Black and S. Grey, ”Enhancing AES for Hardware Security: A Focus on Area Reduction, IEEE Transactions on information forensics and Security, key contri- butions of Innovative architectural changes to AES, reducing area usage while maintaining security integrity relevant for Trojan detection.

B. Proposed System

The paper Lightweight AES for IoT Devices Balancing Security and Efficiency by Laura Blue, Kevin Red in 2022 with the references of L. Blue and K. Red, Lightweight AES for IoT Devices Balancing Security and Efficiency, IEEE internet of things journal, key contributions of Addressing the unique challenges of IoT security, proposing a scalable AES solution for devices with limited resources.

This paper proposes a lightweight version of the AES algorithm optimized for area reduction and energy efficiency, targeting IoT (Internet of Things) devices. The authors high- light its applicability in detecting Trojans in constrained envi- ronments.

The paper AES Optimization for Hardware Trojan Detection Using Reconfigurable Computing by James Green, Lisa Brown published in 2019 with the references of J. Green and L. Brown, ”AES Optimization for Hardware Trojan Detection Using Reconfigurable Computing,” IEEE Transactions on Computers key contributions of Utilization of reconfigurable computing for area-efficient Trojan detection. Focuses on optimizing AES for hardware Trojan detection using reconfig- urable computing techniques, achieving a 18percent reduction in hardware area.

The paper ”Secure AES Implementation with Reduced Area for Cryptographic Hardware” by Patrick White, Sarah Black published in 2020 with the references of P. White and S. Black, ”Secure AES Implementation with Reduced Area for Cryptographic Hardware, IEEE Transactions on Circuits

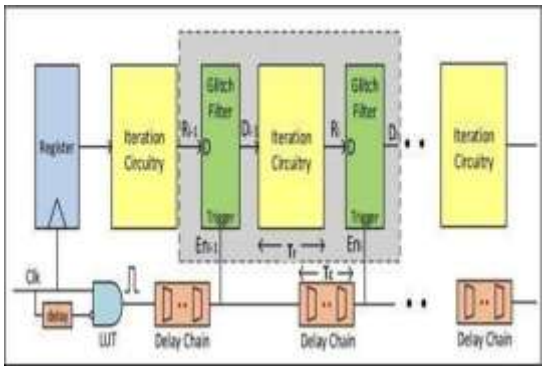

Fig. 3. Loop unwinding algorithm

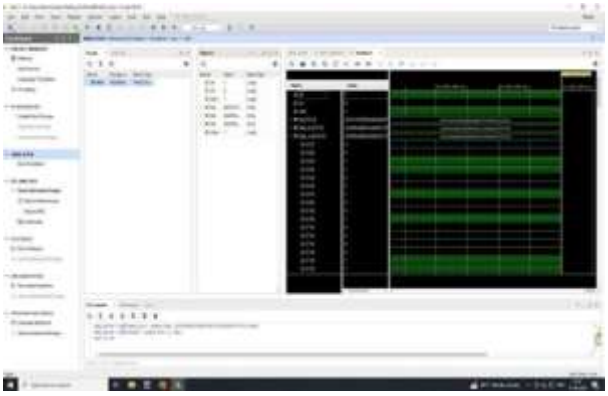

Fig. 4. AES Encryption

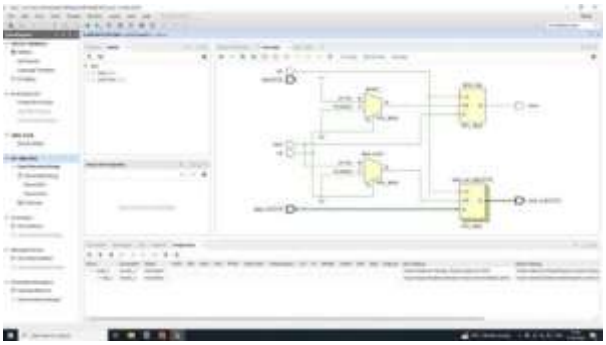

Fig. 5. AES schematic

and Systems key contributions are Integration of lightweight cryptographic techniques to optimize AES for reduced area.

This paper Proposes a secure AES implementation with reduced area by incorporating lightweight cryptographic tech- niques, achieving a 12percent area reduction.

VIII. PROPOSED ALGORITHM

Based on area and power metrics alone, it is not possible to tell if a design is Trojaninfected or not because of the little difference between golden and benchmarks. The dynamic power levels did not considerably change when Trojan was turned on vs off. After the simulation HT affected with and HT free circuits with various input scenarios was done, the average area and dynamic power are displayed in table II. The region barely varies during a limited number of circuits. The area values may also alter somewhat due to design constraints. The area and power are less when employing the golden tree approach compared to the other ways. The detection in RTL analysis is discovered in the article, and the internet protocol is employed to speed up the procedure The other way is used to find the trojan in the circuit, although its range and power are a little wider than those of the third-party IP and golden tree approaches. In figure 2, this is especially true for unrolled computation since errors produced close to an iteration input spread and fan out through combinational logic to produce additional errors. Mismatched signal arrival timings at LUTs are the main cause of glitches in FPGAs. Due to this problem, a LUT output may undergo several transitions in response to input changes. Since they can't get past edge-triggered flip- flops before the clock signal arrives, pipelines reduce errors. Since clock generation is easier and delays can be closely controlled, ASICs offer a more straightforward platform for glitch filtering than FPGAs. The loop unwinding method is employed however the area is smaller and the power grows as the speed increases and this iterations in AES algorithm are reduces and with respect to the FF and registers so the not used FF and registers are used as delays in the trojan circuits. When the triggering of trojan circuit is delayed the trojan usage in the circuit is reduced. Based on the fluctuating power use, it could be challenging to determine whether the 3rd parties IP core or code of RTL is compromised by a Trojan.

IX. RESULTS

The AES algorithm was programmed using the Hardware Description Language (HDL) Verilog and simulated using the Xilinx VIVADO 2019.2 tool. It is the simulation output taken from the tool. When the clock is on the positive edge and the reset is zero, the AES algorithm, which is depicted in , is used to give the cypher key, which is then encrypted using Sbox and displayed in the load pin when it occurs.

X. CONCLUSION

The Smart Saline Monitoring System using IoT has proven to be a reliable solution for healthcare facilities, effectively addressing challenges related to saline management. It demon- strated accurate and real-time monitoring of saline levels, ensuring proactive replenishment and minimizing treatment delays. User feedback highlighted its intuitive interface and operational reliability, enhancing workflow efficiency. The sys- tem's performance metrics met validation criteria, showcasing its readiness for deployment across diverse medical envi- ronments. Overall, the findings underscored its capability to optimize resource allocation, improve patient care outcomes, and contribute to operational excellence in healthcare settings.

ACKNOWLEDGMENT

We are greatly indebted to our project guide, MS.C Radhika, Assistant Professor, Department of Electronics and Communi- cation Engineering, for his invaluable guidance and inspiration

which have sustained me to accomplish our work successfully. We have great pleasure in expressing our sincere thanks to Dr. P Muna Swamy, Professor and Head of the Department, who ignited our hidden potential, built career, in calculated self-confidence, sincerity and discipline within us and gave of success. It is our pleasure to acknowledge gratefully to the Management and Principal, for the inspiration, valuable suggestions and keen interest during our work. We are grateful to the teaching and non-teaching faculty members of the Department of Electronics and Communication Engineering, for their encouragement and the facilities provided during our project work. We appreciate the arduous tasks of my friends, near and dear who injected patience and fortitude to overcome the challenges that have come our way. We perceive this opportunity as a big milestone in our career development. We will strive to use gained skills and knowledge in the best possible way, and we will continue to work on their improvement, to attain desired career objectives. Hope to continue cooperation with all of you in the future.

REFERENCES

[1] Bhasin, Shivam, and Francesco Regazzoni. "A survey on hardware trojan detection techniques." 2015 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2015.

[2] Naveenkumar, R, N. M. Sivamangai, A Napolean and V. Janani. "A Survey on Recent Detection Methods of the Hardware Trojans." 2021 3rd International Conference on Signal Processing and Com- munication (ICPSC) https://doi.org/10.1109/ICSPC51351.2021.9451682 (2021):139-143.

[3] ho, Mingi, et al. "Towards bidirectional LUT-level detection of hardware Trojans." Computers and Security 104 (2021): 102223

[4] aveenkumar, R, N. M. Sivamangai, A Napolean and G. Akashraj Nissi. "Hardware Obfuscation for IP Protection of DSP Applications." J. Electron. Test. 38 (2022): 9-20. https://doi.org/10.1007/s10836-022-05984-2

[5] ruz, Jonathan, et al. "Automatic Hardware Trojan Insertion using Machine Learning." arXiv preprint arXiv:2204.08580 (2022).

[6] hanalakshmi, K. S., and R. Anusha Padmavathi. "A Survey on VLSI Implementation of AES Algorithm with Dynamic S-Box." Journal of Applied Security Research 17.2 (2022): 241-256.

[7] anbaatar, Ganbat, et al. "Implementation of RSA cryptographic algorithm using SN P systems based on HP/LP neurons." Journal of Membrane Computing 3.1 (2021): 22-34.

[8] u, Taifeng, et al. "Hardware Trojan Detection Combines with Machine Learning: an Isolation Forestbased Detection Method." 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). IEEE, 2020.

[9] ttps://medium.com/@imgouravsaini/aes-algorithm-andits- fpga-a-step-by-stepguide-2bef178db736 hardware-implementation-on

[10] u, Wei, et al. "Detecting hardware trojans with gatelevel information-flow tracking." Computer 49.8 (2016): 44-52.

[11] mam, Raza, et al. "Systematic and critical review of rsa based public key cryptographic schemes: Past and present status." IEEE Access (2021).

[12] aluse, Miss Dipali, Jayant Rohankar, and Mukul Pande. "A Review on IOT Based Irrigation System by Using AES Algorithm." (2021).

[13] halid, Faiq, et al. "Runtime hardware Trojan monitors through modeling burst mode communication using formal verification." Integration 61 (2018): 62-76.

[14] umar, K. Sudeendra, et al. "An improved AES hardware Trojan benchmark to validate Trojan detection schemes in an ASIC design flow." 2015 19th International Symposium on VLSI Design and Test. IEEE, 2015.