iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XII **Month of publication:** December 2024

DOI: https://doi.org/10.22214/ijraset.2024.65696

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XII Dec 2024- Available at www.ijraset.com

Design of Malicious Hardware Trojans in AES Crypto system

Sai Kumar Marri
Dept. of Electrical Engineering, The University of Texas at Dallas

Abstract: ***Hardware Trojans pose a significant threat to the security and integrity of cryptographic systems, particularly in Advanced Encryption Standard (AES) implementations, which are widely used in securing sensitive data. These malicious modifications to integrated circuits (ICs) can compromise the confidentiality and reliability of AES cryptographic operations by introducing covert backdoors, information leakage channels, or functional disruptions. Hardware Trojans are typically designed to evade detection during design-time validation and post-manufacturing testing, often activating only under specific triggers such as rare input patterns or environmental conditions.***

In AES systems, Trojans can manipulate the encryption process by leaking secret keys through side-channel information such as power consumption, timing variations, or electromagnetic emissions. Some Trojans directly alter the encryption algorithm, weakening the cryptographic strength and rendering encrypted data vulnerable to attacks. Attackers may also embed Trojans at the Register Transfer Level (RTL) or gate-level design, leveraging the inherent complexity of AES circuits to conceal their presence.

Detection and mitigation of hardware Trojans in AES implementations are challenging due to their stealthy nature. Techniques such as side-channel analysis, functional verification, and static code analysis have been developed, but sophisticated Trojans often bypass these methods. Advanced countermeasures include runtime monitoring, hardware obfuscation, and Trojan-resilient design methodologies.

This paper explores the implications of hardware Trojans in AES cryptographic systems, analysing their design, potential attack vectors, and impacts on data security. Furthermore, it discusses state-of-the-art detection and prevention techniques, highlighting gaps and future research directions. Given the critical role of AES in securing financial, military, and consumer data, addressing the hardware Trojan threat is paramount to ensuring trust in cryptographic hardware and safeguarding against adversarial exploitation.

Keywords: Hardware Trojans, AES Cryptographic Systems, Side-Channel Attacks, Trojan Detection, Cryptographic Security

I. INTRODUCTION

As digital systems permeate nearly every aspect of modern life, ensuring the integrity and security of these systems has become a critical concern. Among the most pressing threats are hardware Trojans—malicious modifications intentionally inserted into integrated circuits (ICs) or hardware designs. These stealthy additions are designed to compromise system functionality, leak sensitive information [3], or create vulnerabilities that attackers can exploit. The rise of globalized semiconductor supply chains and the growing reliance on third-party intellectual property (IP) have heightened the risk of hardware Trojans, making them a formidable challenge in secure system design. One of the most prominent applications affected by hardware Trojans is cryptographic systems, particularly implementations of the Advanced Encryption Standard (AES) [7]. AES is a symmetric encryption algorithm widely used in securing sensitive communications, data storage, and financial transactions. Its resilience to traditional cryptographic attacks has established it as the de facto standard for encryption. However, while AES is robust against mathematical and brute-force attacks, its hardware implementations remain vulnerable to malicious modifications and side-channel attacks [2]. Hardware Trojans embedded in AES circuits can compromise its security by leaking secret keys, altering its functionality, or introducing vulnerabilities that are nearly impossible to detect through traditional testing methods.

Side-channel analysis (SCA) is a critical avenue for both attackers and defenders in the context of hardware security [6]. It involves exploiting unintentional emissions from hardware, such as power consumption, timing information, or electromagnetic radiation, to infer internal states or secret data. Attackers may leverage side-channel analysis in conjunction with hardware Trojans to covertly extract AES encryption keys or decrypt sensitive information. For example, a Trojan might subtly amplify power variations or introduce timing discrepancies that facilitate side-channel attacks [8]. Conversely, researchers and engineers use side-channel analysis techniques to detect anomalies and identify Trojan activity, underscoring its dual role in hardware security.

Detecting hardware Trojans is a particularly challenging task due to their stealthy design. Attackers deliberately craft Trojans to evade traditional testing and validation processes. For instance, Trojans may remain dormant during functional testing and activate only under rare conditions or specific input patterns [9]. This behaviour significantly complicates detection efforts, as exhaustive testing of all possible states and inputs is impractical for complex hardware systems. Furthermore, the subtle modifications introduced by Trojans often mimic benign design features or natural hardware faults, further obscuring their presence.

State-of-the-art Trojan detection methods can be broadly classified into pre-silicon and post-silicon techniques. Pre-silicon methods include formal verification, static code analysis, and design-for-trust approaches, which aim to ensure hardware integrity during the design phase [5]. These methods focus on identifying unused or weakly connected logic, analysing control flow, and validating functional properties. Post-silicon techniques, on the other hand, rely on physical testing and side-channel fingerprinting to detect anomalies in fabricated chips. These include power analysis, timing analysis, and electromagnetic scans to identify discrepancies indicative of malicious circuits. Despite advancements, existing detection methodologies face significant limitations in scalability, effectiveness, and cost, especially for highly integrated and sophisticated designs [10].

In the context of AES cryptographic systems, the stakes are particularly high. Hardware Trojans targeting AES implementations can have severe consequences, compromising the confidentiality and integrity of sensitive data. For example, Trojans can leak encryption keys through covert channels, introduce weaknesses in the encryption algorithm, or degrade system performance to enable timing-based attacks [11]. The widespread use of AES in critical applications such as secure communications, financial systems, and military operations amplifies the impact of these vulnerabilities, necessitating robust detection and prevention strategies. The interplay between hardware Trojans, AES cryptographic systems, and side-channel analysis underscores the complexity of ensuring hardware security in modern digital systems. While significant progress has been made in understanding the threat landscape and developing countermeasures, the dynamic and evolving nature of these attacks demands continuous innovation [4]. This paper delves into the nexus of these domains, exploring the design, implications, and detection of hardware Trojans in AES implementations. It examines the use of side-channel analysis both as a tool for attackers to exploit vulnerabilities and for defenders to detect malicious modifications. Finally, it surveys state-of-the-art detection techniques, highlighting their strengths, limitations, and potential avenues for future research [12].

By addressing the challenges posed by hardware Trojans in AES cryptographic systems, this paper aims to contribute to the broader effort of securing hardware against malicious adversaries [1]. The need for robust and scalable solutions is paramount as hardware security becomes an increasingly critical component of the global cybersecurity landscape. Through a comprehensive analysis of the threats, methodologies, and emerging trends, this paper seeks to provide valuable insights for researchers, practitioners, and policymakers striving to build trust in modern computing systems document is a template. For questions on paper guidelines, please contact us via e-mail.

II. HARDWARE TROJANS

This paper outlines various types of hardware Trojans specifically designed to compromise AES cryptographic systems, detailing their triggers, payloads, and the affected files. Below is a summary of the Trojans described in the document:

1) Trojan 1: Transmitting Plain Text on the Terminal

Trigger: Pressing ini_system and start encryption buttons simultaneously.
Payload: Displays unencrypted plaintext on the terminal.

2) Trojan 2: Transmitting Key with Cipher Text

Trigger: Pulling all switches high (FF) and pressing the start_tx button.
Payload: Leaks encryption keys alongside the cipher text.

3) Trojan 3: Leaking Key via LEDs

Trigger: Holding the ini_system button and pressing start_tx periodically.
Payload: Displays encryption key bit-by-bit on LEDs.

4) Trojan 4: Power Profile Modification

Trigger: Always active, no specific trigger required.
Payload: Alters the power profile to leak key information.

5) Trojan 5: Denial of Service (DoS)
Trigger: Setting the key select to its inverse value.
Payload: Halts system operations, displaying "00" on the terminal.

6) Trojan 6: DoS via Keyboard Input
Trigger: Pressing SHIFT+F4 during encryption.
Payload: Freezes the entire system.

7) Trojan 7: Corrupting the System
Trigger: Pressing ENTER key, and later SHIFT+q.
Payload: Turns off LEDs, confusing users about system state.

8) Trojan 8: DoS via Button Press
Trigger: Pressing ini_system and start_tx simultaneously.
Payload: Freezes the system completely.

9) Trojan 9: Leaking Key on Monitor
Trigger: Pressing the ini_system button thrice.
Payload: Displays the master encryption key on the monitor.
Each Trojan demonstrates a unique way to exploit AES systems, highlighting vulnerabilities that can compromise confidentiality, integrity, and availability. If you'd like, I can provide more in-depth analysis or explore countermeasures for these threats.
FPGA board used for implementing and testing hardware Trojans in an AES crypto system. Xilinx Spartan-6 on Basys-2 board is used for the implementation of AES crypto system along with the Malicious trojans and provide basic functions:

- Basic Input/Output Functions: Buttons and switches (ini_system, start_tx, etc.) are used to trigger operations. LEDs are used to display information, such as encryption keys.
- Cryptographic Processing: It can run AES encryption and managing UART communication.
- Resource Utilization Analysis: Indicates that the FPGA has sufficient computational and hardware resources for modifying and executing hardware Trojans.

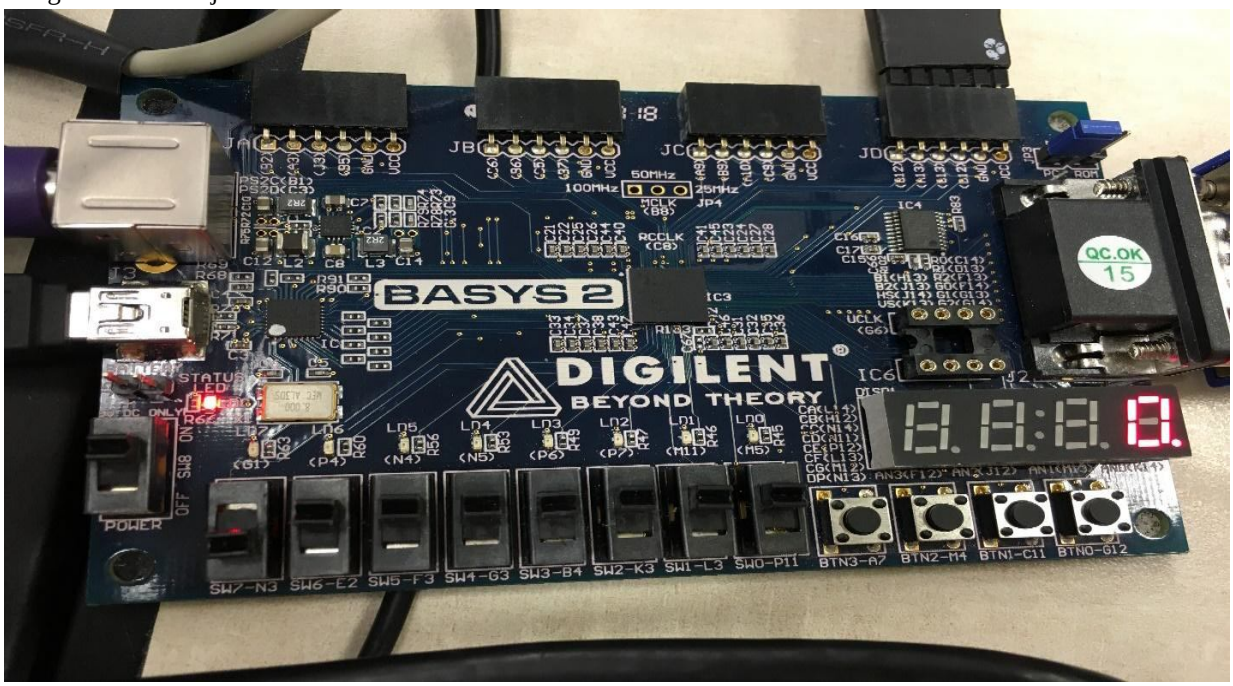


Fig. 1: Xilinx Spartan6 on Basys2 board

A. Transmitting Plain Text on the Terminal

In this Trojan, any text entered by the user through the keyboard, i.e., the plaintext, is displayed directly on the terminal without being encrypted by the encryption algorithm. This occurs after the ini_system button is pressed on the FPGA. While holding the ini_system button, the user simultaneously presses the start encryption button, triggering the plaintext to appear on the terminal instead of being processed into ciphertext.

Trigger: ini_system and encryption start button on the FPGA.

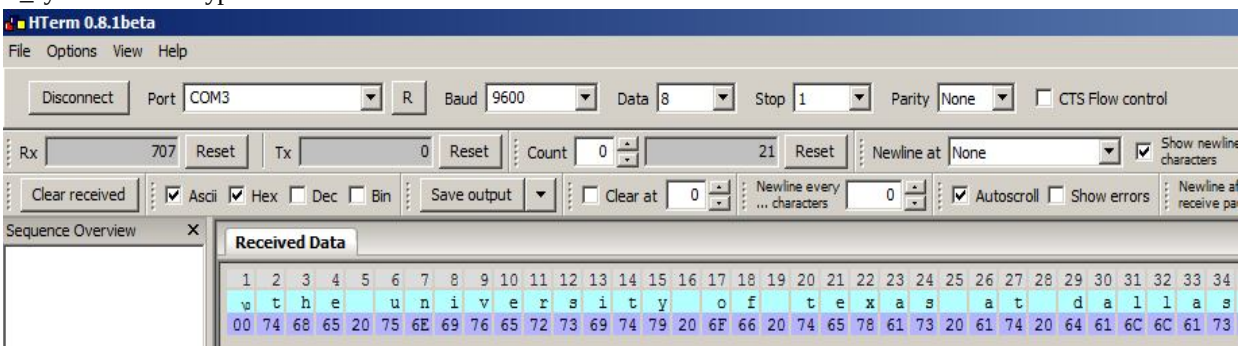


Fig. 2: Demonstration of Trojan leaking the sensitive data

B. Transmitting Key with Cipher Text

In this Trojan, After the ini_system button is pressed on the FPGA the plain text is entered by the user using the keyboard and before the encryption start button is pressed on the FPGA, all the switches on the board are pulled up which is it is made FF.The start_tx button is pressed and the cipher text gets displayed on the terminal along with the key used for encrypting.

Trigger: All switches on the board are pulled high (FF).

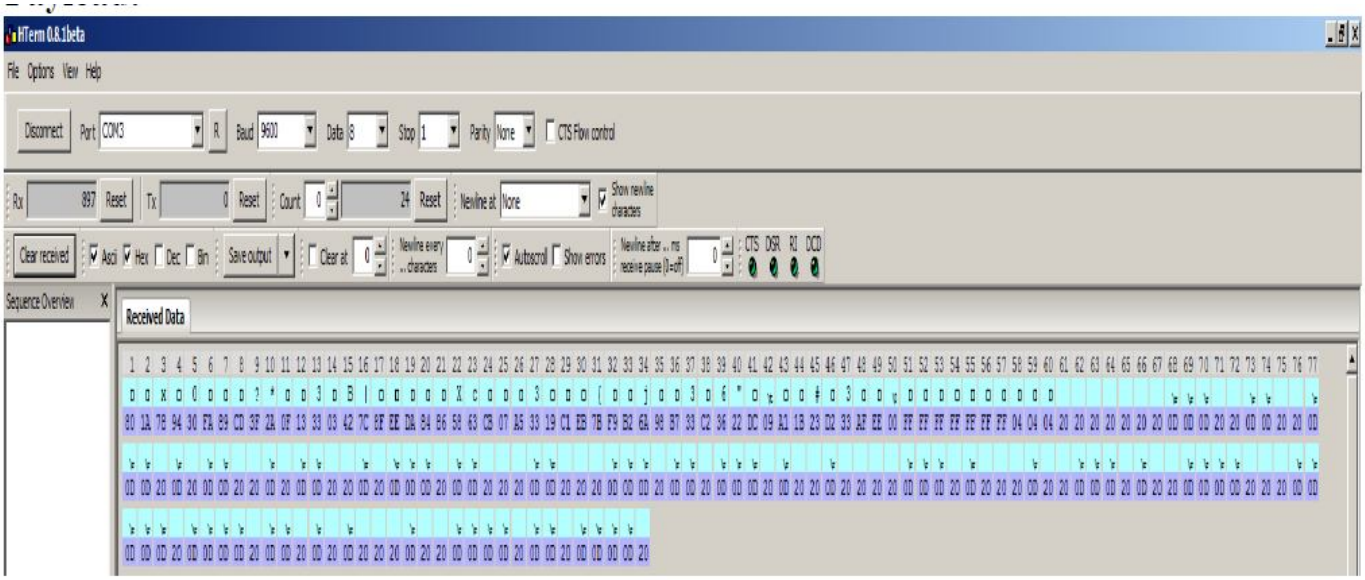


Fig. 3: Demonstration of leaked data from system

C. Leaking the Key on the LEDs after Encryption

In this Trojan, the plain text is entered by the user after ini_system button is pressed .The encryption takes place after start encryption button is pressed on the FPGA .The encrypted data gets transmitted after the start_tx button is pressed which displays the cipher text .After the cipher text is viewed and in order to leak the key on the LED's ,the ini_system button has to be pressed continuously and the start transmission button is also pressed for every 8 bits of the key .Thus all the 128 bits of the key can be leaked on the LED's.

Trigger: The ini_system button has to be pressed continuously, and the start transmission button is also pressed for every 8 bits of the key.

D. Modifying power profile to leak Key

In this Trojan, we are modifying the power profile by adding another signal latched from the clock. This Latched signal will be functional during the UART transmission. Latched clock signal will be gated/un-gated based on the key bits. During the first byte transmission on UART latched signal either toggle to leak lsb key bit specifying as '1' or latched signal gets gated to leak lsb key bit as '0'.

Trigger: No trigger (always ON)

E. Denial of Service (DoS)

In this Trojan, Denial of Service takes place .The ini_system button is pressed on the board with the key select value being 10000000(SW8 SW7 SW6 SW5 SW4 SW3 SW2 SW1 SW0) and the plain text is entered and then the start encryption button is pressed on the board making the switches on the board the inverse of the key select value that is 0111111 which makes the entire system to stop its normal operation and displays only "00"on the terminal.

Trigger: key select=~key select

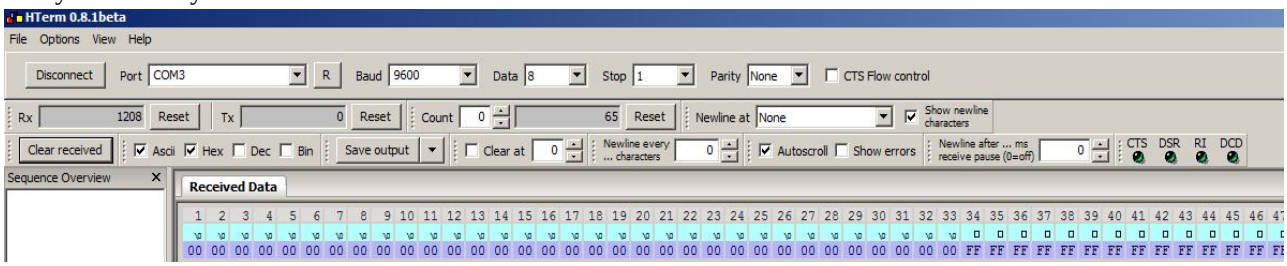


Fig. 4: Demonstration of leaked keys

F. DoS via Keyboard Input

In this Trojan also the payload is Denial of Service. In order for the user to start encryption of the plain text, the ini_system is first pressed followed by start encryption button and whenever "SHIFT+F4" is pressed on the keyboard the entire system freezes.

Trigger: SHIFT+F4

G. Corrupting the System

In this Trojan, the ini_system button is pressed on the board and if the ENTER key is pressed on the keyboard and if the user enters the plain text the LED's of the FIFO buffer is switched off which thereby confuses the user if his plain text is entered and if that plain text will be encrypted and if "SHIFT+q" is pressed the normal operation takes place and the LED's are switched on.

Trigger: ENTER key

H. DoS via Button Press

In this Trojan, the payload is Denial of Service. The ini_system button is pressed which enables the user to enter his plain text and once when the ini_system button and start_tx button is pressed entire system freezes and the Denial of Service takes place.

Trigger: ini_system and start_tx button is pressed simultaneously

I. Leaking Key on Monitor

In this Trojan, the plain text is entered by the user after the ini_system button is pressed thrice. The encryption takes place after start encryption button is pressed on the FPGA. The encrypted data gets transmitted after the start_tx button is pressed which displays the cipher text on the monitor along with the master key.

Trigger: ini_system button is pressed thrice

III.CONCLUSIONS

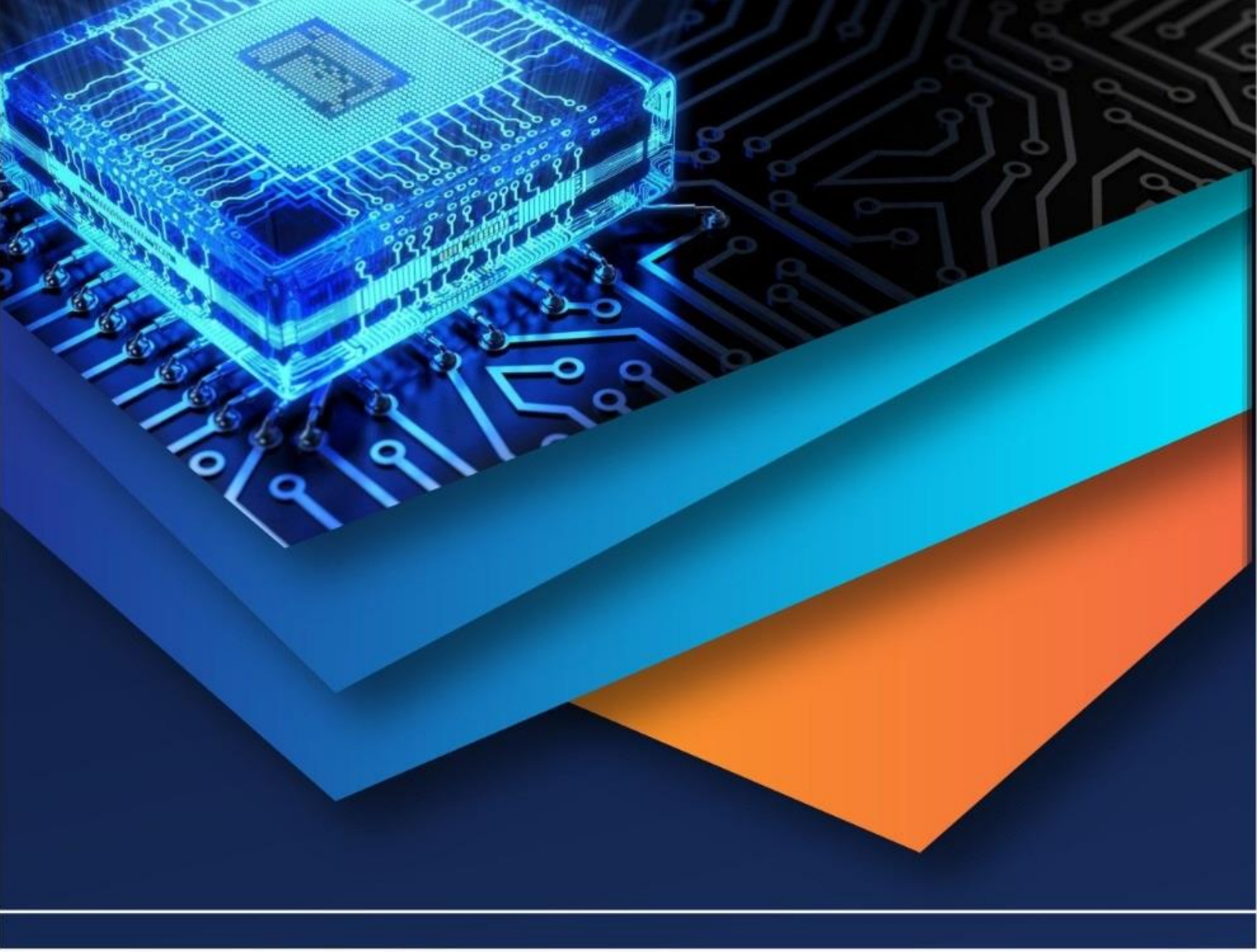
The threat of hardware Trojans in AES cryptographic systems underscores the critical importance of securing hardware against malicious modifications. AES, as a cornerstone of modern encryption, is widely deployed across sensitive applications, from financial systems to military communications. While the algorithm itself is robust against cryptographic attacks, its hardware implementations remain vulnerable to stealthy Trojans designed to leak secret keys, compromise encryption integrity, or facilitate side-channel exploitation [13].

These vulnerabilities can lead to catastrophic breaches, undermining trust in secure systems. Detecting and mitigating hardware Trojans in AES systems is a formidable challenge due to their ability to evade traditional testing and validation processes. Trojans are often designed to remain dormant during functional testing and activate under rare or specific conditions, making their detection through standard methodologies impractical. Moreover, their subtle modifications are engineered to blend into complex hardware designs, complicating efforts to distinguish malicious circuits from benign ones [14].

Emerging detection techniques, such as side-channel fingerprinting, formal verification, and runtime monitoring, offer promising avenues for identifying and countering Trojans in AES implementations. However, these approaches are not without limitations, including scalability challenges, high computational costs, and vulnerability to sophisticated attack designs. The dynamic evolution of Trojan strategies necessitates ongoing research and innovation to stay ahead of adversaries [15]. As hardware security becomes increasingly critical in the global cybersecurity landscape, securing AES systems against hardware Trojans demands a multidisciplinary approach. This includes advancements in detection methodologies, the integration of Trojan-resilient design practices, and a deeper understanding of side-channel behaviors. By addressing these challenges, the industry can enhance the resilience of AES implementations, safeguarding sensitive data and ensuring the reliability of cryptographic hardware. The fight against hardware Trojans is not merely a technical endeavor but a foundational effort to maintain trust in secure systems in an increasingly interconnected world.

REFERENCES

[1] Tehranipoor, M., & Koushanfar, F. (2010). "A Survey of Hardware Trojan Taxonomy and Detection." IEEE Design & Test of Computers, 27(1), 10-25.
[2] Guin, U., Huang, K., DiMase, D., Carulli, J. M., Tehranipoor, M., & Makris, Y. (2014). "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain." Proceedings of the IEEE, 102(8), 1207-1228.
[3] Salmani, H., Tehranipoor, M., & Plusquellic, J. (2012). "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(1), 112-125.
[4] Xu, X., Shakya, B., Tehranipoor, M., & Forte, D. (2017). "Novel Bypass Attack and BDD-based Tradeoff Analysis Against all Known Logic Locking Attacks." In Cryptographic Hardware and Embedded Systems – CHES 2017, 189-210.
[5] Xiao, K., & Tehranipoor, M. (2013). "BISA: Built-In Self-Authentication for Preventing Hardware Trojan Insertion." In Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 45-50.
[6] Nahiyan, A., Xiao, K., Yang, K., Jin, Y., Forte, D., & Tehranipoor, M. (2016). "AVFSM: A Framework for Identifying and Mitigating Vulnerabilities in FSMs." In Proceedings of the 53rd Annual Design Automation Conference (DAC), 1-6.
[7] Guin, U., Shi, Q., Forte, D., & Tehranipoor, M. (2016). "FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs." ACM Transactions on Design Automation of Electronic Systems (TODAES), 21(4), 63.
[8] Rahman, M. T., Rahman, M. S., Wang, H., Tajik, S., Khalil, W., Farahmandi, F., Forte, D., Asadi, N., & Tehranipoor, M. (2020). "Defense-in-Depth: A Recipe for Logic Locking to Prevail." Integration, the VLSI Journal, 72, 39-57.
[9] Waksman, A., Suozzo, M., & Sethumadhavan, S. (2013). "FANCI: Identification of Stealthy Malicious Logic Using Boolean Functional Analysis." In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 697-708.
[10] Sturton, C., Hicks, M., Wagner, D., & King, S. T. (2011). "Defeating UCI: Building Stealthy and Malicious Hardware." In Proceedings of the 2011 IEEE Symposium on Security and Privacy, 64-77.
[11] Sai Kumar Marri, N. Muthiah. "Obscure Hardware Trojan Design in 8051 Micro-controller". International Journal of Modern Engineering Research 14. 06(2024): 43-49.
[12] Hicks, M., Finnicum, M., King, S. T., Martin, M. M. K., & Smith, J. M. (2010). "Overcoming an Untrusted Computing Base: Detecting and Removing Malicious Hardware Automatically." In Proceedings of the 2010 IEEE Symposium on Security and Privacy, 159-172.
[13] Khalid, F., Hasan, S. R., Hasan, O., & Shafique, M. (2018). "SIMCom: Statistical Sniffing of Inter-Module Communications for Run-time Hardware Trojan Detection." arXiv preprint arXiv:1901.07299.
[14] Srivastava, A., Das, S., Choudhury, N., Psiakis, R., Silva, P. H., Pal, D., & Basu, K. (2023). "SCAR: Power Side-Channel Analysis at RTL-Level." arXiv preprint arXiv:2310.06257.
[15] Bursztein, E., Invernizzi, L., Král, K., Moghimi, D., Picod, J.-M., & Zhang, M. (2023). "Generalized Power Attacks against Crypto Hardware using Long-Range Deep Learning." arXiv preprint arXiv:2306.07249.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY