ESP-IJCEET
ESP International Journal of Communication Engineering & Electronics Technology
ISSN: 2583-9217 / Volume 1 Issue 3 October 2023 / Page No: 16-22
Paper Id: IJCEET-V1I3P103 / Doi: 10.56472/25839217/ IJCEET-V1I3P103

Original Article

An Efficient Cube Stripper Based Logic Locking Technique for Digital Circuits

M.Sowmiya[1], D.Ajitha[2]

[1,2]*SRM Madurai College for Engineering and Technology, Tamilnadu, India*

Received Date: 09 September 2023 Revised Date: 22 September 2023 Accepted Date: 04 October 2023

Abstract: *The continuous evolution of integrated circuits and their deployment in critical applications demand robust security mechanisms to safeguard sensitive data and intellectual property. Logic locking techniques have emerged as a prominent approach to protect circuits from unauthorized access and intellectual property theft. However, the increasing sophistication of attacks against these logic locking mechanisms necessitates innovative defenses. This project introduces a novel Cube Stripping-based Functional Analysis Attack (FALL attack) targeting state-of-the-art Logic Locking algorithms. We present the methodology and results of our research, highlighting the success of our proposed approach in breaching the security of circuits locked with Cube Strip Secure Function Logic Locking (SFLL), currently recognized as the only combinational locking algorithm resilient to all known attacks.*

Keywords: *Cube Stripper, Digital Circuits, Locking Technique.*

I. INTRODUCTION

The proliferation of integrated circuits (ICs) across various domains, including aerospace, automotive, and telecommunications, has brought about unprecedented benefits but has also exposed these circuits to potential security threats. Protecting the confidentiality and integrity of the designs, algorithms, and sensitive data embedded within these ICs has become paramount. Logic locking, a widely adopted technique in the field of hardware security, offers a means to fortify the security of ICs by encrypting their functionality, thereby safeguarding them against reverse engineering and intellectual property theft.

Over the years, logic locking has witnessed significant advancements, with researchers and industry practitioners striving to create algorithms capable of withstanding a spectrum of attacks. Among these attacks, Functional Analysis Attacks (FAAs) have emerged as a potent threat, focusing on identifying the secret key used in locking by analyzing the functionality of the locked circuit. In response to this challenge, we present a new approach known as Cube Stripping-based Functional Analysis Attacks (FALL attacks) aimed at circumventing the formidable security defenses provided by the state-of-the-art Logic Locking algorithms.

In this paper, we present the findings of our research, outlining the methodologies employed in FALL attacks and the experimental results obtained. We pay special attention to the efficacy of our proposed method against circuits secured using Cube Strip Secure Function Logic Locking (SFLL), an algorithm acknowledged for its resilience against all known attacks. Our contributions shed light on the evolving landscape of hardware security and provide insights into the development of more robust and resilient logic locking techniques.

II. LITERATURE REVIEW

Limaye et al. have introduced a defense mechanism called DisORC, aimed at detecting any unauthorized access to a chip's scan interface. This defense system promptly erases all traces of the encryption key and disconnects it from the circuit, rendering scan chains unusable while the correct key is not in the system. Additionally, DisORC enhances the security of TRLL by protecting it from oracle-guided attacks.

In a different approach, Juretus et al. have presented CORALL, a method that enhances security against SAT attacks for modified logic cones requiring substantial corruption of primary circuit outputs. This quantified corruption is compared between an activated and locked state of the integrated circuit. Moreover, CORALL's modifications to logic cones improve resilience against structural attacks.

Chiang et al. have proposed a cyclic logic locking technique designed to thwart state-of-the-art attacking methods. Experimental results indicate that this method effectively locks general combinational benchmarks with minimal area overhead, promising advancements in logic locking techniques.

Zhou et al. have introduced a new design for logic-locking blocks that differ from the traditional clustered '1'-cells in K-maps. Instead, they employ checkboard patterns of '1'-cells. While this approach necessitates AND functions with numerous inputs to resist SAT attacks, it offers the advantage of combining many such AND functions with XOR gates.

Rathor et al. have presented a lightweight logic locking technique that neutralizes sensitization and cone-based attacks effectively. They also propose new multi-key input gates and a replacement-based logic locking algorithm, which not only prevents sensitization attacks but also eliminates rare-triggered nets from the design.

Tsai et al. have introduced a dominant gate algorithm to achieve a high security standard for logic locking. This method simultaneously manages to exceed a 50% Hamming Distance (HD) while remaining exempt from key sensitization, showcasing comprehensive defense ability.

Rezaei et al. have offered a novel perspective on logic encryption by integrating locking and obfuscation for a sensitive component of a circuit. They highlight that the security impact of this encryption can extend to the entire circuit as long as the critical component's Confidentiality Ratio (CR) and Availability Ratio (AR) remain high.

Xiang-Min Yang et al. have conducted an analysis of LOOPLock's locking mechanism and introduced an attacking approach based on the analysis of the locking structure. To counter this new attack, they propose LOOPLock 2.0, which strengthens the original cyclic logic locking method. Experimental results demonstrate the efficiency and effectiveness of this approach.

Ayush Jain et al. have proposed a novel differential fault analysis (DFA) attack based on stuck-at faults. This attack can be used to break logic locking relying on a stored secret key. It leverages self-referencing by injecting faults into key lines and comparing the response with the fault-free counterpart.

Sirone et al. have presented a phase mask inscription technique utilizing two-beam interferometry to create nonhomogeneous period gratings known as chirped fiber Bragg gratings. Inscription experiments with deep ultraviolet excimer and femtosecond laser sources illustrate how this method depends on the coherence properties of the inscription laser.

Karousos et al. have proposed a weighted logic locking technique that employs multiple key-inputs to control each key-gate, making it immune to key-sensitization attacks. This approach achieves a 50% Hamming Distance (HD) even for circuits with multiple outputs.

Chakraborty et al. have surveyed the evolution of logic locking over the past decade, elucidating various aspects of this field and its applications, such as processor pipelines, GPUs, and analog circuits. Their aim is to serve as a primer for researchers interested in developing new logic locking techniques.

Chan-Hong Park et al. argue that corruptibility for incorrect keys is a vital metric for logic locking. They describe an ATPG-based method to measure corruptibility, especially for large circuits. Results from applying this method to various circuits demonstrate its effectiveness.

Shamsi et al. have proposed an approximate SAT-based attack framework that focuses on iterative convergence, reducing a compound scheme to a standalone SAT-resilient scheme. They also introduce a novel technique to increase the corruptibility of SAT-resilient protection schemes in a controlled manner.

Nejat et al. have introduced a new logic locking method using MUX cells instead of traditional XOR/XNOR key-gates. This approach offers benefits like increased switching activity and design functionality obfuscation.

Usui et al. have proposed an adaptive locking technique that dynamically determines whether a critical section should be executed transactionally or while holding a mutex lock. Their approach includes adaptivity logic, cost-benefit analysis, and low-overhead statistics collection in a full C compiler.

De-Xuan Ji et al. have introduced a glitch-based logic locking method designed for sequential circuits. This method generates glitches and employs rising and falling transitions as key-inputs for comprehensive logic locking.

In a novel development, Chakraborty et al. have introduced the Structural Functional (SURF) attack, which accomplishes key extraction through scalable functional analysis while leveraging the output of structural attacks. They have developed a complete flow and an automatic tool for this attack, yielding promising results.

III. PROPOSED SYSTEM

A significant breakthrough in defeating all the existing logic locking techniques was the Boolean satisfiability (SAT)-based key-pruning attack, commonly known as the SAT attack. This attack is centered around the concept of eliminating incorrect keys by utilizing distinguishing input patterns (DIPs). DIPs are derived by constructing a miter circuit that involves two instances of the locked netlist; these two circuits share the same primary inputs but use different key inputs. A DIP is identified when the outputs of these two copies of the locked netlist differ. To carry out this process, a functional integrated circuit (IC) with the secret key loaded in its memory acts as an oracle to iteratively pinpoint the incorrect keys.

The computational complexity of the SAT attack is measured in terms of the number of DIPs generated during the attack. Recent research endeavors in the field of logic locking have primarily concentrated on devising defenses against the SAT attack, recognizing its potency and significance in compromising security.

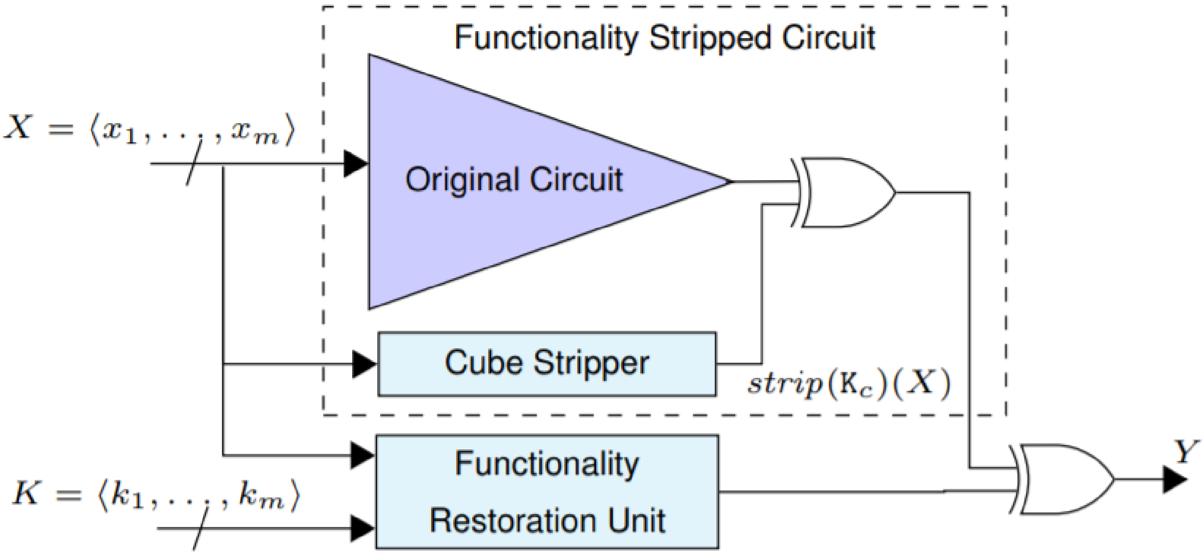
Functionality Stripped Circuit
$X = \langle x_1, \ldots, x_m \rangle$
Original Circuit
Cube Stripper
$strip(\mathtt{K}_c)(X)$
Functionality
Restoration Unit
$K = \langle k_1, \ldots, k_m \rangle$
Y

Figure 1: Proposed System

Much subsequent work has focused on SAT-attack resilient logic locking that ensures the number of equivalence classes of keys is exponential in the key length. Broadly speaking, these proposals all share the structure shown in Figure. They introduce a circuit which “flips” the output refer to this component as the cube stripping unit. This flipped output is then inverted by a key-dependent circuit that we refer to as the progammable functionality restoration unit. This latter circuit is guaranteed to have an exponential number of equivalence classes of keys and ensures SAT attack resilience. (Note these methods “hard code” the locking key in the cube stripping unit which leads to a vulnerability that we exploit.) Initial proposals along these lines were Anti-SAT] and SARLock. However, Anti-SAT was vulnerable to the signal probability skew (SPS) attack while SARLock was vulnerable to the Double DIP attack and the Approximate SAT attack. Both schemes are vulnerable to removal and bypass attacks . Subsequently, Yasin et al. proposed TTLock and Secure Function Logic Locking (SFLL) To the best of our knowledge, SFLL is the only combinational logic locking scheme resilient to all of the above attacks.

A. Ring Oscillator (RO) PUF:

The concept of Physically Unclonable Functions (PUF) was initially introduced by Pappu in 2001. Since then, numerous PUF designs have been proposed. PUFs can be categorized into two main types based on their sources of randomness: extrinsic PUFs and intrinsic PUFs. Extrinsic PUFs, such as Optical PUF and Coating PUF, introduce variations manually and explicitly during the manufacturing process. On the other hand, intrinsic PUFs, which are more prevalent, rely on natural randomness arising from parameter deviations and mechanical mismatches.

Among intrinsic PUFs, the Arbiter PUF (APUF) was one of the first silicon PUFs, utilizing an arbiter to compare the delay of two identical paths. However, achieving symmetry in routing, especially in field-programmable gate arrays (FPGAs), can be challenging, leading to issues with uniqueness. To alleviate routing challenges, the Ring Oscillator (RO) PUF emerged as an FPGA-friendly design, comparing oscillating periods instead of single path delays. Nonetheless, RO PUFs offer fewer response bits than APUFs of the same area. More response bits, known as Challenge-Response Pairs (CRPs), translate to a longer service lifetime in terms of authentication cycles. In other words, RO PUFs need more area to collect the same number of CRPs as APUFs.

Chen et al. introduced an even-stage RO PUF named Bistable Ring PUF, which compares the rising path delay with the falling path delay. However, this design may require a significant amount of time for the responses to become stable. In addition to delay-based PUFs, memory-based PUFs represent another important category, using existing on-chip memory to generate identifiers. Static Random Access Memory (SRAM) PUF is a typical example, relying on the uncertain initial state during power-up as a source of randomness. Nevertheless, it can be costly to power down and up when authentication is required during normal operation. To address this issue, Butterfly PUF and Buskeeper PUF replace the memory unit with D flip-flops and buskeepers, respectively.

Furthermore, other memory-based PUFs like Memristor PUF, Magnetic RAM PUF, and Resistance RAM PUF have been proposed. However, most memory-based PUFs exhibit a linear relationship between the number of CRPs and the amount of memory units. Despite the presentation of various PUF structures, they all face two common challenges: unreliability and predictability. Unreliability stems from the sensitivity of device parameters to environmental factors, such as temperature and voltage supply, making it difficult to keep all CRPs unchanged. Error correction code (ECC) techniques, such as BCH codes or fuzzy extractors, and fault-tolerant techniques, like majority voting, are often employed to enhance reliability. Predictability is primarily attributed to correlations between CRPs, associated with specific hardware structures or unaccounted measurement dependencies. For instance, APUFs' path delay can be accurately approximated using a linear additive model, making them susceptible to machine learning attacks. Among all silicon intrinsic PUFs, RO PUF stands out as a promising choice, offering a balanced tradeoff between various metrics.

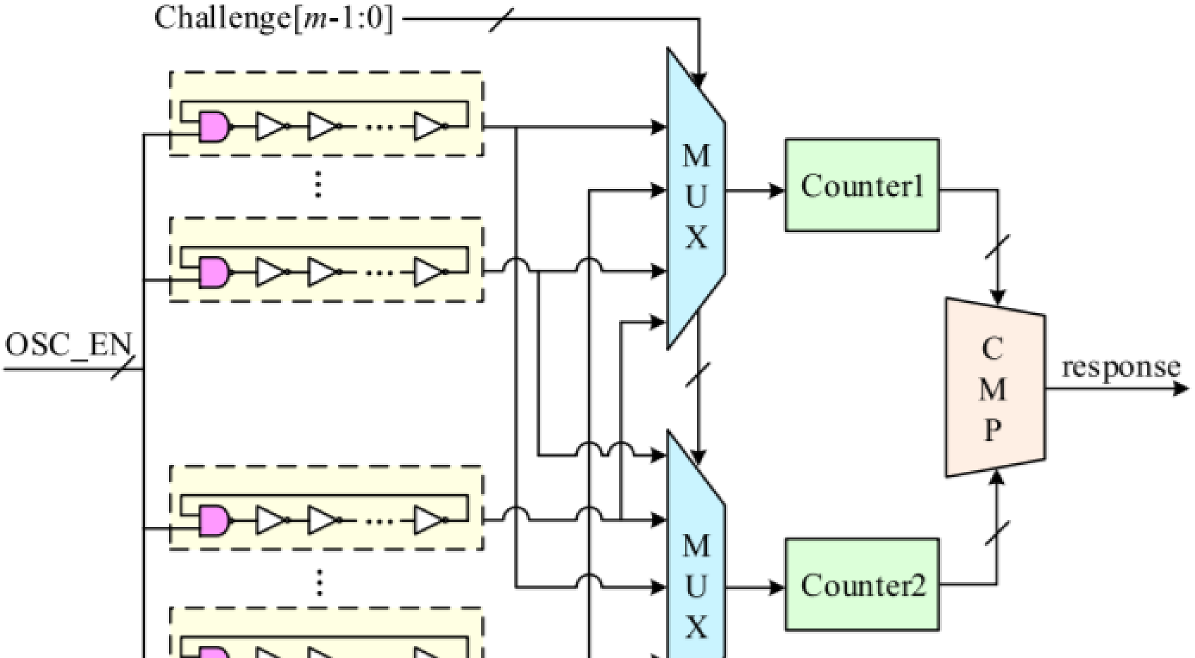

Figure 2: Structure of Traditional RO PUF

As depicted in Figure 1, the traditional Ring Oscillator (RO) Physically Unclonable Function (PUF) comprises N ROs (Ring Oscillators), two N-to-1 multiplexers, two counters, and one comparator. Each RO is constructed using a NAND gate and an even number of inverters.

The challenge, represented by an m-bit value, selects two distinct ROs through the multiplexers. Once the oscillation enable (OSC_EN) signal is activated, the chosen ROs begin oscillating, and they drive the subsequent counter to tally the number of oscillation cycles. After a defined period, denoted as t (also known as the measuring period), the OSC_EN signal is deactivated, causing all ROs to cease oscillating simultaneously.

Subsequently, a comparator is employed to compare the count values from the two counters. Despite the structural similarity of all ROs, their frequencies differ due to manufacturing variations. Consequently, if the top RO oscillates at a higher frequency, the comparator outputs a 1 as the response bit; otherwise, it outputs a 0. By repeating this process with n distinct challenges applied in a similar manner, a n-bit response can be generated.

IV. IMPLEMENTATION RESULTS

After performing the synthesize process, the RTL schematic has been created automatically based on the functionality. The routing between the different cells can be viewed clearly by this schematic.

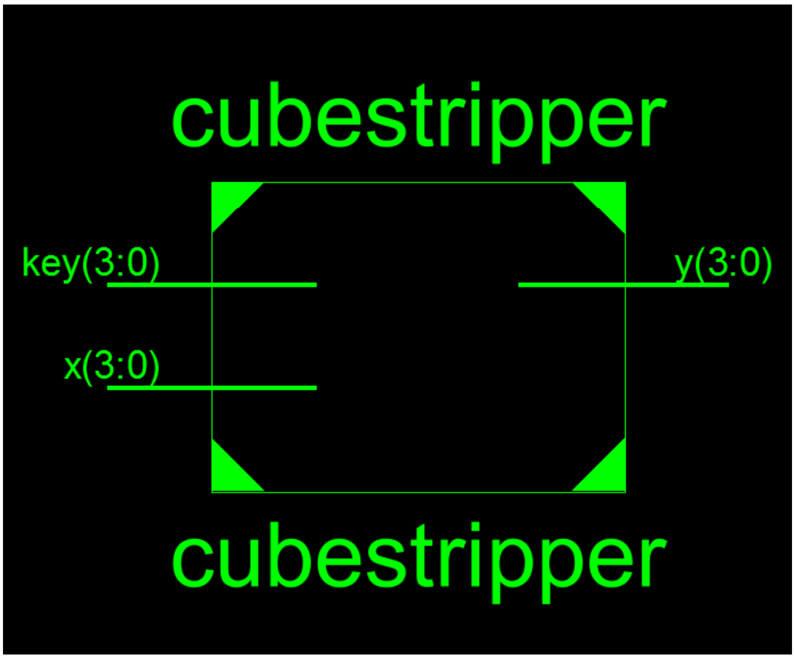

Figure 3: RTL Schematic

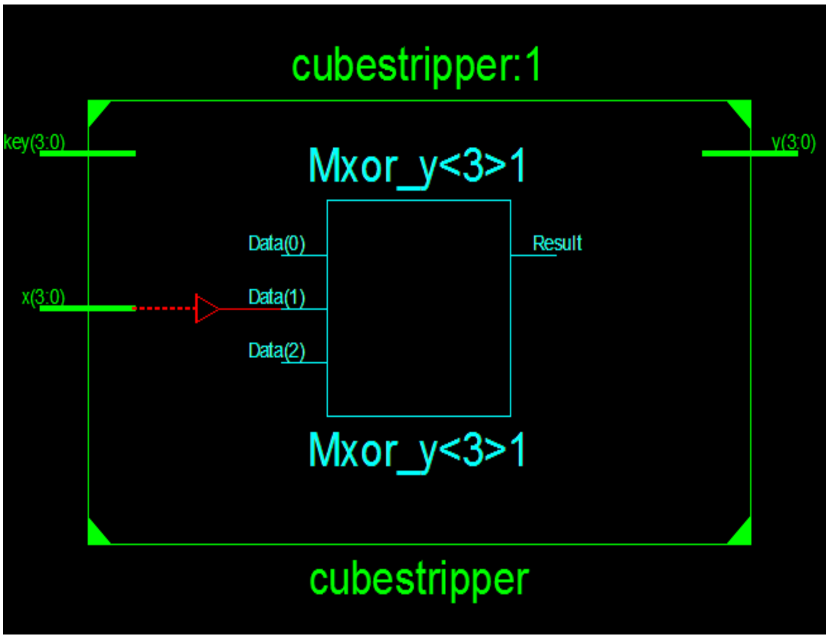

Figure 4: Gatelevel Netlist

V. PERFORMANCE ANALYSIS

The presented figure below illustrates a substantial reduction in both time and area based on the implementation results achieved using the Spartan-3 processor. The proposed algorithm demonstrates a remarkable reduction in area consumption in comparison to the current system.

Table 1: Performance Analysis

Benchmark		slices	LUT	IOB	delay(ns)
c432	conventional	61	108	43	24.813
	modified	61	108	49	24.018
	Stripper	63	108	44	24.813
C880	conventional	63	108	86	19.503
	modified	67	115	92	18.27
	Stripper	63	109	87	19.503
C1355	conventional	45	78	73	12.499
	modified	47	84	79	12.703
	Stripper	44	79	74	12.950
C2670	conventional	84	160	373	15.094
	modified	85	162	373	21.938
	Stripper	85	162	374	21.938

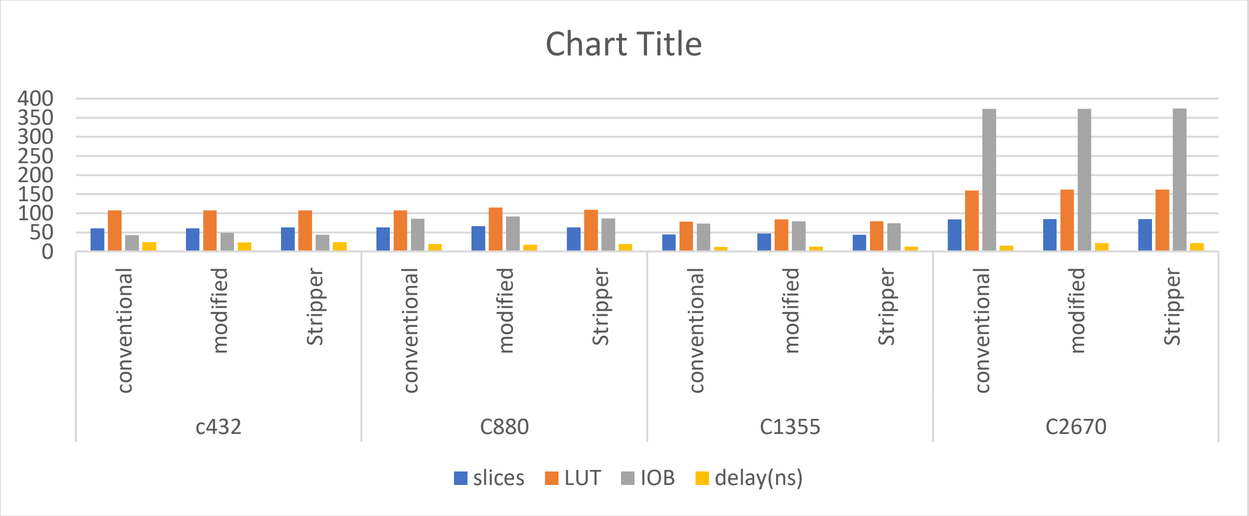

Figure 5: Performance Analysis Chart

The table presents a comparative analysis of different implementations across various benchmark cases. These implementations are categorized as "conventional," "modified," and "Stripper," and they are evaluated based on several key metrics.

Firstly, the "Slices" metric indicates the number of slices used in each implementation. Slices are fundamental components in FPGA designs and encompass resources like lookup tables (LUTs) and flip-flops. This metric provides insights into the resource utilization of each configuration.

The "LUT" column specifies the count of lookup tables employed in each implementation. Lookup tables are versatile building blocks that can be programmed to execute specific logic functions. Their usage showcases the complexity of the logic being implemented.

Additionally, the "IOB" metric quantifies the number of Input/Output Blocks used. IOBs play a crucial role in handling input and output signals in FPGA designs. Monitoring their usage is essential for assessing the I/O capabilities of each configuration.

Finally, the "delay(ns)" column presents the propagation delay in nanoseconds for each benchmark case. This metric measures the time it takes for a signal to travel through the implemented logic. Lower delay values generally indicate faster performance.

In summary, this table offers a comprehensive view of how different implementations stack up in terms of resource utilization (slices, LUTs, IOBs) and signal propagation delay. It provides valuable insights into the efficiency and performance of each configuration in the context of the specified benchmarks.

VI. CONCLUSION

In this project, a novel approach was introduced, focusing on Cube Stripping-based Functional Analysis Attacks, commonly abbreviated as FALL attacks, which target state-of-the-art Logic Locking algorithms. FALL attacks leverage both structural and functional analyses of locked circuits to uncover the encryption key used for locking. During experimental testing, our newly proposed method demonstrated a high degree of success in breaking the security of existing systems, particularly benchmark circuits that had been secured using the Cube Strip Secure Function Logic Locking (SFLL) method. It's worth noting that SFLL is currently recognized as the sole combinational locking algorithm capable of withstanding all known attacks.

VII. REFERENCES

[1] Chakraborty, Prabuddha; Cruz, Jonathan; Bhunia, Swarup (2019). [IEEE 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) - McLean, VA, USA (2019.5.5-2019.5.10)] 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) - SURF: Joint Structural Functional Attack on Logic Locking. , (), 181–190

[2] Ji, De-Xuan; Chiang, Hsiao-Yu; Lin, Chia-Chun; Wu, Chia-Cheng; Chen, Yung-Chih; Wang, Chun-Yao (2019). [IEEE 2019 32nd IEEE International System-on-Chip Conference (SOCC) - Singapore (2019.9.3-2019.9.6)] 2019 32nd IEEE International System-on-Chip Conference (SOCC) - A Glitch Key-Gate for Logic Locking. , (), 74–79.

[3] Usui, Takayuki; Behrends, Reimer; Evans, Jacob; Smaragdakis, Yannis (2009). [IEEE 2009 18th International Conference on Parallel Architectures and Compilation Techniques (PACT) - Raleigh, North Carolina, USA (2009.09.12-2009.09.16)] 2009 18th International Conference on Parallel Architectures and Compilation Techniques - Adaptive Locks: Combining Transactions and Locks for Efficient Concurrency. , (), 3–14.

[4] Nejat, Arash; Kazemi, Zahra; Beroulle, Vincent; Hely, David; Fazeli, Mahdi (2019). [IEEE 2019 IEEE 4th International Verification and Security Workshop (IVSW) - Rhodes Island, Greece (2019.7.1-2019.7.3)] 2019 IEEE 4th International Verification and Security Workshop (IVSW) - Restricting Switching Activity Using Logic Locking to Improve Power Analysis-Based Trojan Detection. , (), 49–54.

[5] Shamsi, Kaveh; Meade, Travis; Li, Meng; Z.Pan, David; Jin, Yier (2018). On the Approximation Resiliency of Logic Locking and IC Camouflaging Schemes. IEEE Transactions on Information Forensics and Security, (), 1–1.

[6] Chan-Hong Park, ; Jin Wook Kim, ; Beomsup Kim, (2000). [IEEE Second IEEE Asia Pacific Conference on ASICs AP-ASIC 2000 - Cheju, South Korea (28-30 Aug. 2000)] Proceedings of Second IEEE Asia Pacific Conference on ASICs. AP-ASIC 2000 (Cat. No.00EX434) - A 1.8-GHz self-calibrated phase-locked loop with precise I/Q matching. , (),

[7] Chakraborty, Abhishek; Jayasankaran, Nithyashankari Gummidipoondi; Liu, Yuntao; Rajendran, Jeyavijayan; Sinanoglu, Ozgur; Srivastava, Ankur; Xie, Yang; Yasin, Muhammad; Zuzak, Michael (2019). Keynote: A Disquisition on Logic Locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (), 1–1.

[8] Karousos, Nikolaos; Pexaras, Konstantinos; Karybali, Irene G.; Kalligeros, Emmanouil (2017). [IEEE 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS) - Thessaloniki, Greece (2017.7.3-2017.7.5)] 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS) - Weighted logic locking: A new approach for IC piracy protection. , (), 221–226.

[9] Sirone, Deepak; Subramanyan, Pramod (2020). Functional Analysis Attacks on Logic Locking. IEEE Transactions on Information Forensics and Security, 15(), 2514–2527.

[10] Ayush Jain;M Tanjidur Rahman;Ujjwal Guin; (2020). ATPG-Guided Fault Injection Attacks on Logic Locking . 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE), (), –.

[11] Xiang-Min Yang;Pei-Pei Chen;Hsiao-Yu Chiang;Chia-Chun Lin;Yung-Chih Chen;Chun-Yao Wang; (2022). LOOPLock 2.0: An Enhanced Cyclic Logic Locking Approach . IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (), –.

[12] Rezaei, Amin; Shen, Yuanqi; Zhou, Hai (2020). [IEEE 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) - Grenoble, France (2020.3.9-2020.3.13)] 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) - Rescuing Logic Encryption in Post-SAT Era by Locking & Obfuscation. , (), 13–18.

[13] Tsai, I-Chun; Liu, Fang-Ru; Feng, Jianhua (2019). [IEEE 2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC) - Xi'an, China (2019.6.12-2019.6.14)] 2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC) - A Dominant Gate Insertion Algorithm Implementation for Logic Locking in IP Protection. , (), 1–3.

[14] Rathor, Vijaypal Singh; Sharma, G. K. (2019). A Lightweight Robust Logic Locking Technique to Thwart Sensitization and Cone Based Attacks. IEEE Transactions on Emerging Topics in Computing, (), 1–

[15] Zhou, Jingbo; Zhang, Xinmiao (2020). [IEEE 2020 IEEE International Symposium on Circuits and Systems (ISCAS) - Sevilla (2020.10.12-2020.10.14)] 2020 IEEE International Symposium on Circuits and Systems (ISCAS) - A New Logic-Locking Scheme Resilient to Gate Removal Attack. , (), 1–5.

[16] Chiang, Hsiao-Yu; Chen, Yung-Chih; Ji, De-Xuan; Yang, Xiang-Min; Lin, Chia-Chun; Wang, Chun-Yao (2019). LOOPLock: LOgic OPtimization based Cyclic Logic Locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (), 1–1.

[17] Juretus, Kyle; Savidis, Ioannis (2020). Increased Output Corruption and Structural Attack Resiliency for SAT Attack Secure Logic Locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (), 1–1.

[18] Limaye, Nimisha; Kalligeros, Emmanouil; Karousos, Nikolaos; Karybali, Irene G.; Sinanoglu, Ozgur (2020). Thwarting All Logic Locking Attacks: Dishonest Oracle with Truly Random Logic Locking. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (), 1–1.