Dynamically Obfuscated Scan Chain To Resist Oracle-Guided Attacks On Logic Locked Design

M Sazadur Rahman, Adib Nahiyan, Sarah Amir, Fahim Rahman,
Farimah Farahmandi, Domenic Forte, Mark Tehranipoor
Email: {mohammad.rahman,sarah.amir,adib1991,fahim034}@ufl.edu,{farimah,dforte,tehranipoor}@ece.ufl.edu

I. Introduction

Logic obfuscation inserts additional key gates in the design to hide original functionality of the design in order to prevent from reverse engineering and re-suing. The proper function of the design is ensured once correct unlocking key inputs are provided along with the original inputs. The design remains locked throughout the different phases of the supply chain. Without knowing the correct secret key the correct functionality cannot be retrieved. Once finished ICs are delivered to a trusted facility, the unlocking key inputs are driven by an on-chip tamper-proof memory and the design gets activated. While logic obfuscation can be an effective technique to establish trust among different entities of the IC supply chain, it has not seen application due to its lack of attack resiliency. For example, most logic obfuscation techniques are vulnerable to *Boolean satisfiability (SAT)*-based oracle-guided attack. SAT attack [1] breaks most combinational logic obfuscation techniques in a matter of short-time by finding distinguishing input patterns (DIPs) to rule out incorrect keys utilizing the output corruptibility of the miter circuit constructed using locked design and activated design. For sequential designs, it is assumed that an IC's internal states can be accessed and controlled via scan chains to read/write the value of the flip-flops. To resist SAT attack, several SAT-resistant logic obfuscation techniques have been proposed-SARLock [2], Anti-SAT [3] and SFLL [4]. SARLock and Anti-SAT resists SAT attack by increasing the number of required distinguishing input patterns (DIPs) exploiting a one-point function to corrupt the output of the design for all the incorrect keys. While these two SAT resistant techniques are strong enough to withstand the power oracle-guided attack but are vulnerable to Bypass attack [5], SPS attack [6], AppSAT [7] attack. SFLL [4] technique strips some of the functionality of the original design and hides it in the form of a secret key. Once correct secret key is applied, original functionality of the design is restored. For long SFLL has been the state-of-the-art SAT resistant logic obfuscation technique. Very recently a new functional analysis attack (FALL) [8] has been proposed that uses structural and functional analyses of locked design to identify the vulnerability of cube stripping circuit.

From Table I, it can be seen that none of the existing countermeasures can provide full protection against all types of attacks. For example, most countermeasures targeting scan-based side-channel attacks, does not make consideration of protecting against IP Piracy, over-production, tampering and counterfeiting. On the other hand, SAT-resistant logic obfuscation techniques [2]–[4] are sometimes strong enough to protect SAT attack [1] but collapses when confronts Bypass attack [5], SPS attack [6], AppSAT [7] attack and very recently FALL attack [8]. To address the above-mentioned security shortcomings, this article adopts a novel dynamically-obfuscated scan (DOS) design [19], performs its complete security assessment against SAT attack and other scan side-channel attacks. The advantages of dynamically-obfuscated scan (mentioned as DOSC throughout this literature) design are as follows:

- It can protect IPs against existing non-invasive scan-based attacks while maintaining the testability and pattern application flexibility.
- It can protect IPs against exiting *satisfiability*-based SAT attacks and its variants.
- It offers low area/power overhead, has negligible impact on industrial design and test flow, and there is no increase in test time.

The rest of this article is organized as follows: Section II describes the proposed architecture. The DOSC-based implementation and test methodology is presented in Section III. SAT attack analysis in DOSC inserted functional obfuscated circuit is discussed in Sec IV.

II. DOSC Architecture

The details of DOSC architecture and how it secures a design is discussed here. The DOSC architecture reads $Control\ Vector$ from non-volatile directly memory access (DMA) in secure zone, and provides protection to scan chains. The DOSC architecture has capacity and flexibility to provide protection for IP owner as well as IC integrator. IP owner can either integrate one DOSC into IP, as IP core II, or share the central DOSC belonging to the customized logic, as IP core I. As illustrated in Figure 1, the proposed DOSC architecture is composed of a *linear feedback shift register (LFSR)*, a *Shadow Chain* with XOR gates, and a *Control Unit*.

A. Details of DOSC

1) LFSR: The polynomial primitive LFSR is adopted to generate a λ-bit *Obfuscation Key* (λ is the length of scan chains), which is used to scramble scan in/out vectors as shown in Figure 1. The *Obfuscation Key* is protected through the AND gates of the Shadow Chain. The LFSR is being driven by the Control Unit, and changes its output based on the permutation rate α, where

$$\alpha = \frac{Scan\ Clock\ Frequency}{Obfuscation\ Key\ Update\ Frequency} \qquad (1)$$

TABLE I: Different Oracle-Guided Attacks, Countermeasures and their Vulnerabilities

Attacks	**Assets**	**Existing Countermeasures**	**Vulnerabilities of Existing Countermeasures**
Differential Attack [9]	Internal States	Test Mode Protection [10]–[13]	Test mode only attack [14]
Resetting Attack [15]	Internal Secrets	LCSS [16], Lock & Key [17], Scan Interface Encryption [18]	Bit-role Identification Attack [19]
Flushing Attack [20]	Internal Secrets		
Combinational Function Recovery [21]	Functionality	DOSC [19]	
SAT Attack [1]	*Obfuscation Key*	SARLock [2], Anti-SAT [3], SFLL [4]	Bypass Attack [5], SPS Attack [6], FALL Attack [8]
SMT Attack [22]		N/A	N/A
ScanSAT [23]		N/A	N/A
AppSAT [7]		SFLL [4]	FALL Attack [8]
SPS Attack [6]	SAT Resistant Techniques		
Bypass Attack [5]			

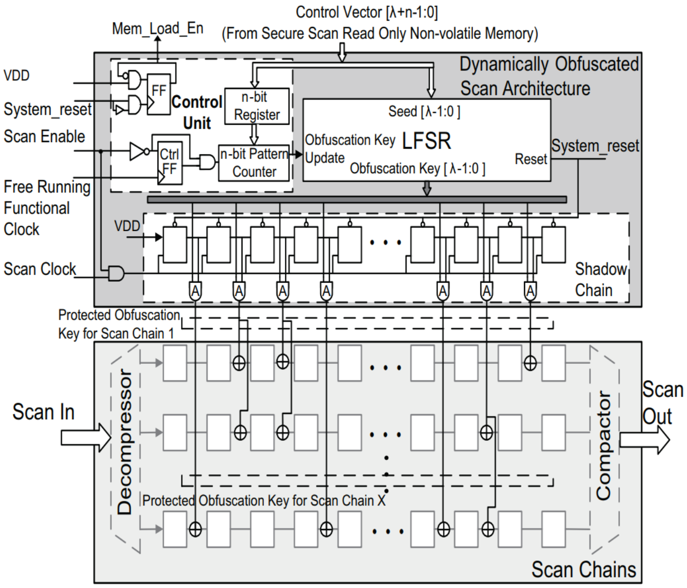

Fig. 1: Detailed architecture of the adopted DOSC [19].

It should be noted that for LFSR, a seed with all zeros (ones) is illegal when using an XOR (XNOR) feedback.

2) Shadow Chain and XOR Gates: As shown in Figure 1, the input of the Shadow Chain is the λ-bit *Obfuscation Key* generated by the LFSR, while the outputs are λ-bit *Protected Obfuscation Key*s. The Shadow Chain is designed for propagating the *Obfuscation Key* at the i-th scan cell along the scan chain when the i-th scan clock comes. Therefore, the Shadow Chain is able to i) protect the *Obfuscation Key* from being leaked through resetting attack, ii) prevent any unscrambled data from being scanned out, iii) prevent adversaries from scanning in values intentionally, and at the same time, make no impact on structural and chain tests. It can be seen that the Shadow Chain is designed as a cascade of λ flip-flops, which is driven by the scan clock gated by scan enable signal. As shown in Figure 1, the data input of its first flip-flop is connected to VDD. The XOR gate inserted after the i-th scan cell of Scan Chain X is controlled by the output of the i-th flip-flop of the Shadow Chain through a Type A AND gate. As shown in Figure 1, the Type A AND gates of DOSC are the AND gates connecting the scan cells within Shadow Chain, the *Obfuscation Key* bits generated by the LFSR, and the XOR gates inserted into the scan chain, which actually are used to gate the individual *Obfuscation Key* bits by the scan cells of Shadow Chain. After reset, as the scan clock forces the flip-flop along the Shadow Chain to logic 1 one by one, only when the last flip-flop in the Shadow Chain becomes logic 1 at the λ-th scan clock, the scrambled response starts to show up at the scan output. At the same time, the Shadow Chains ith flip-flop starts to obfuscate the ith flip-flop of Scan Chain X at the ith scan clock, which prevents the attacker from scanning in any intended values. Therefore, if the attacker keeps flushing the scan chain, an original or inverted scan in sequence shows up at the scan output after λ bits of zeros. Furthermore, as the *Protected Obfuscation Key* has been settled down after the whole chain is scanned, the Shadow Chain does not impact the DFT launching or capturing process, e.g., when applying stuck-at or transition delay faults. Then the scrambled test responses are scanned out. The Shadow Chain should be synchronously reset with the LFSR at any reset event. As all of the DFT scan chains are scanned synchronously, and the length of the scan chain is usually short with on-chip compression, the architecture only needs one single short Shadow Chain, which has low area penalty. Furthermore, as the Shadow Chain is plugged into the scan chains, it is not bypassable.

3) Control Unit: The Control Unit, as shown in Figure 1, is designed to control memory loading as well as LFSR activities, which is composed of a small n-bit register, a n-bit pattern counter, as well as a control flip-flop. During system initialization, a Control Vector is loaded from the secure scan read-only non-volatile DMA, which includes a λ-bit seed for the LFSR, an n-bit value p determining the *Obfuscation Key* update frequency, and the maximum *Obfuscation Key* update count. The Control Unit of DOSC generates the Mem Load En signal. This signal allows the Control Vector of DOSC to be loaded from DMA once after system reset. The Control Vector is determined by the IC designer. As a part of system firmware, the Control Vector is stored into read only nonvolatile memory located in secure zone with DMA, which satisfies: 1) immediate Control Vector accessing: the Control Vector is automatically loaded into DOSC at powering up, which can be guaranteed by hard coding the Control Vector address in DMA; 2) limited readability: the Control Vector can only be read by DOSC, which can be satisfied by using the handshaking signal Mem Load En (in Figure 1) generated by DOSC, as an input of the DMA address accessing authorization. Additionally, as shown in Figure 1, during scan, Mem_Load_ En also enables the Control Vector can only be read once after the reset event.

DOSC Operation: Based on the three major components introduced above, the obfuscation flow of the proposed design is summarized below. In Step 1, during system initialization, a Control Vector is loaded to the LFSR and the Control Unit, which is composed of a seed for the LFSR and a vector to determine the *Obfuscation Key* update frequency. In Step 2, the

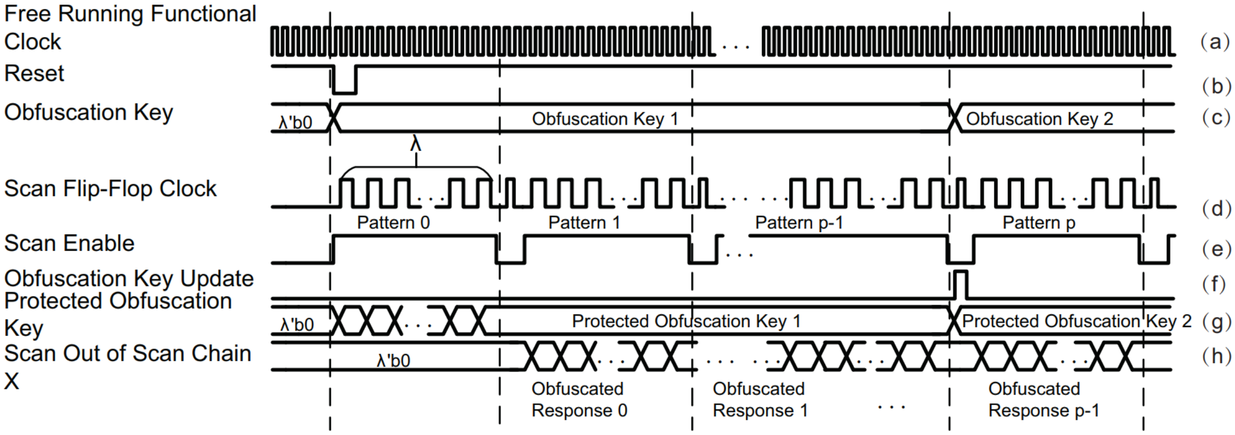

Fig. 2: The timing diagram of DOSC architecture [19].

Obfuscation Key is generated at the output of the LFSR, which is driven by the Control Unit. In Step 3, during the first λ scan clocks after reset, the *Protected Obfuscation Key* is generated bit by bit based on the Shadow Chain and the *Obfuscation Key*. In Step 4, at the λth scan clock, the *Protected Obfuscation Key* settles down. Then, all the test patterns and responses will be scrambled based on the *Protected Obfuscation Key*. Figure 2 shows the timing diagram of the proposed design. It can be seen that the *Obfuscation Key* is generated at the output of the LFSR in waveform (c), and is dynamically changed every p patterns (p is configurable by the IP owner), when the *Obfuscation Key* update is enabled and generated by the Control Unit (waveforms (c) and (f)). As presented before, after reset, the *Protected Obfuscation Key* for Scan Chain X generated by the Shadow Chain is updated bit by bit with the scan clock, and settles down at the λth scan clock (waveform (g)). During the period of the first λ scan clocks, the scan out is locked to 0. Once the λ-th scan clock comes, the scan out starts to output obfuscated responses (waveform (h)).

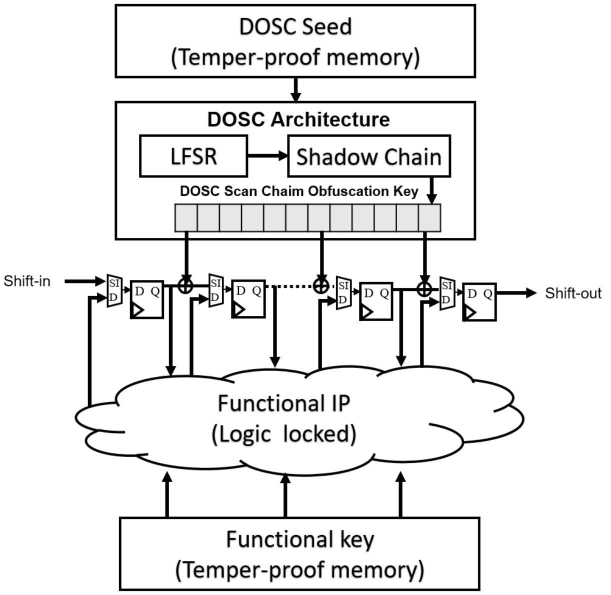

Fig. 3: DOSC inserted in a functional obfuscated benchmark.

B. DOSC inserted benchmark

As shown in Figure 3, DOSC architecture is inserted in the scan chain of a logic locked functional IP. Both the LFSR seed and functional key is maintained with same level of protection in a tamper-proof memory. LFSR of DOSC architecture generates pseudo random numbers based on the DOSC seed. LFSR generated pseudo random numbers are passed through shadow chain to generate DOSC scan chain obfuscation key. XOR gates of DOSC architecture are placed in the scan chain of logic locked functional IP in an uniform manner that ensures maximum security. One of the input to the XOR gates are coming from the scan chain while the other input is coming from the DOSC scan chain obfuscation key. Let us consider the length of scan chain is N. When the design is switched to scan mode, for the first N number of cycles all zero pattern comes out of scan-out port. After N scan clock cycle, scrambled patterns starts coming out of scan-out port. Any pattern shifting though the scan chain will then be scrambled by DOSC generated scan chain obfuscation key. This scrambling process protects - (i) the design from scan-based side channel attack, and (ii) functional key from *Boolean satisfiability*-based SAT attack [1].

III. Implementation and Test methodology

A. Test Methodology

Scan-based tests are required for wafer, assembly, and sometimes system tests. An overview of the test methodology with DOSC is shown in Figure 4.

At IP Owners: As shown in Figure 4, at IP owner, stuck-at, transition, or delay test patterns/responses without obfuscation are generated at first by IP owners. This step can be implemented by using the final DOSC inserted netlist, and forcing *Protected Obfuscation Key* as λb0. Then, according to the predetermined seed, LFSR function, and the location of XOR gates, which are only known by the IP owner, the obfuscated test patterns, and fault-free responses are generated. The obfuscated test patterns and responses will be delivered to testers downstream in supply chain, i.e., IC integrator, foundry, assembly/test facilities.

At Foundry/Assembly: During the first system initialization at foundry, the encrypted Control Vector is programmed into the non-volatile directly memory access (DMA) with other

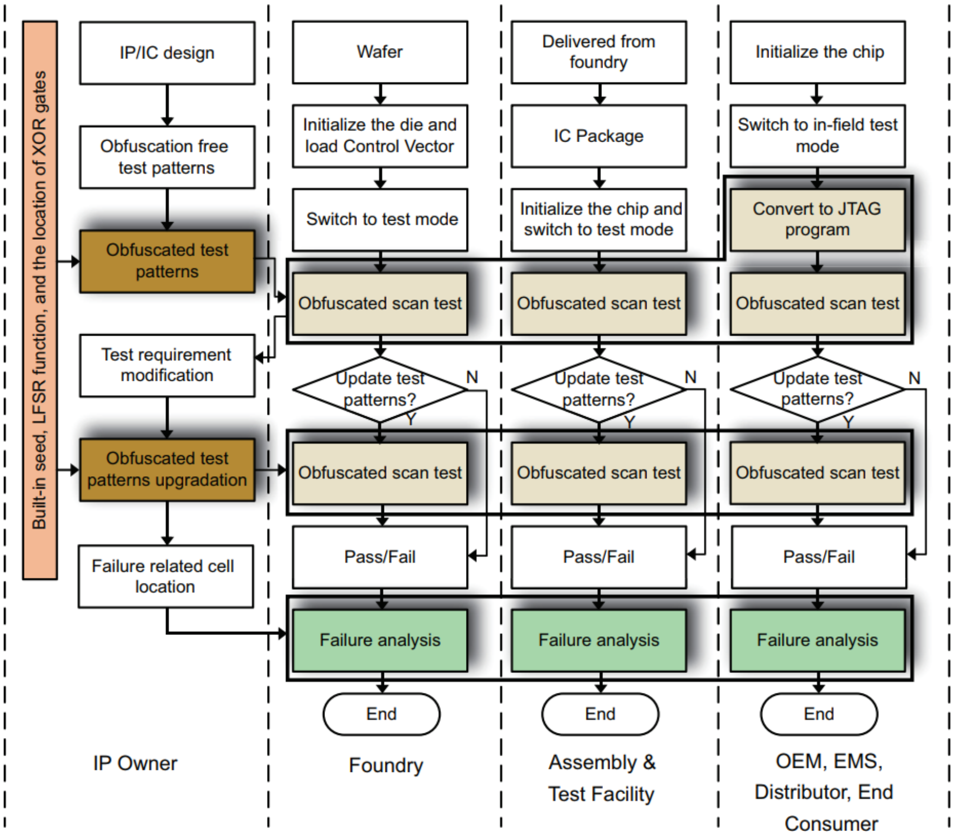

Fig. 4: DOSC based test methodology for different supply chain entities [19].

system configurations, which provides seeds for Obfuscated Key generation at each power up. Then, the chip is ready for testing. During obfuscated scan test, the obfuscated patterns delivered by IP owner are applied to chips, and the obfuscated responses are collected by test engineers at foundry or assembly to detect fault. There is no increase in test time compared with the original scan test. Sometimes, due to test requirement adjustment, test engineer at fab/assembly or IP owner needs to add/remove some test patterns, or reorder the test patterns. According to the adjusted test requirement, test engineer or IP owner can update the obfuscated test patterns/responses with flexibility. By comparing the collected test responses and the fault-free obfuscated responses, the test engineers at fab or assembly can make the pass/fail decision. The failure analysis needs to be assisted by the IP owner. As the obfuscation is bit wise, the failure bits are the same for both obfuscated and plain responses. Thus the IP owner can locate area of defect on the layout using the plain responses, and deliver the area coordinate to the failure analysis facility.

At OEM/EMS/Distributor and End Customer: After the chip is integrated into PCB, the product engineers in OEM, EMS, distributor and end customer may perform scan-based test via data interfaces (i.e., JTAG) for in-field debug. Thus the automatic test equipment (ATE) test patterns need to be converted to satisfy the interface protocol. The converted patterns are then applied to IC under test. Based on the quality of original test patterns, IP owner may update the scrambled test patterns, and fault-free responses. Then the product engineer uses the adjusted scrambled test patterns and responses to perform in-field debug again to maximize inspection test quality. The failure analysis still needs the help of IP owner. The product engineer locates the failed obfuscated response bits and sends the bit index to the IP owner. The IP owner then delivers the defect area coordinate to the failure analysis facility.

IV. SAT Attack Analysis

To perform SAT attack on DOSC inserted functional obfuscated circuit, the attacker needs to unroll the sequential DOSC structure, convert it to a combinational equivalent circuit, then perform SAT attack to reveal the seed of the LFSR. With seed revealed and knowing the expression of the LFSR, the attacker can identify dynamic scan obfuscation key at any cycle which breaks the scan obfuscation security and clears attacker's path to perform SAT attack to find functional obfuscation key. From an attacker's perspective, this attack model should be the most promising case to compromise security of DOSC-integrated obfuscated circuit where regular SAT attack on functional obfuscation key through scan-chain does not work. The attacker can also target functional obfuscation key and perform SAT attack to reveal unlocking key and seed of DOSC at the same time. In this approach functional (obfuscated) circuitry will be an added complexity on top the complexity of breaking DOSC circuitry. In another approach, an attacker can unroll the sequential design and exploit primary IOs to perform a variant of SAT attack using bounded model checking [24]. However, sequential circuit unrolling has an adverse effect on SAT attack complexity. Massad et al. [24] have attempted to break camouflaged sequential circuits which are conceptually similar to obfuscated sequential circuits, without scan access using bounded model checking. However, the authors were only successful for very small sequential design where the number of sequential elements was less that 250. For any industrial scale design such attack is not feasible due to state space explosion [25].

References

[1] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 137–143.

[2] M. Yasin, B. Mazumdar, J. J. Rajendran, and O. Sinanoglu, "Sarlock: Sat attack resistant logic locking," in *Hardware Oriented Security and Trust (HOST), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 236–241.

[3] Y. Xie and A. Srivastava, "Anti-sat: Mitigating sat attack on logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[4] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1601–1618.

[5] X. Xu, B. Shakya, M. M. Tehranipoor, and D. Forte, "Novel bypass attack and bdd-based tradeoff analysis against all known logic locking attacks," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 189–210.

[6] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Security analysis of anti-sat," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 2017, pp. 342–347.

[7] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "Appsat: Approximately deobfuscating integrated circuits," in *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 95–100.

[8] D. Sirone and P. Subramanyan, "Functional analysis attacks on logic locking," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 936–939.

[9] J. D. Rolt, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "A novel differential scan attack on advanced dft structures," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 4, p. 58, 2013.

[10] D. Hely, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Secure scan techniques: a comparison," in *12th IEEE International On-Line Testing Symposium (IOLTS'06)*. IEEE, 2006, pp. 6–pp.

[11] G.-M. Chiu and J. C.-M. Li, "Ieee 1500 compatible secure test wrapper for embedded ip cores," in *2008 IEEE International Test Conference*. IEEE, 2008, pp. 1–1.

[12] L. Pierce and S. Tragoudas, "Enhanced secure architecture for joint action test group systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 7, pp. 1342–1345, 2012.

[13] J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "A smart test controller for scan chains in secure circuits," in *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*. IEEE, 2013, pp. 228–229.

[14] S. S. Ali, O. Sinanoglu, S. M. Saeed, and R. Karri, "New scan-based attack using only the test mode," in *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2013, pp. 234–239.

[15] G. Sengar, D. Mukhopadhyay, and D. R. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 2080–2084, 2007.

[16] J. Lee, M. Tebranipoor, and J. Plusquellic, "A low-cost solution for protecting ips against scan-based side-channel attacks," in *24th IEEE VLSI Test Symposium*. IEEE, 2006, pp. 6–pp.

[17] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing scan design using lock and key technique," in *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*. IEEE, 2005, pp. 51–62.

[18] M. Da Silva, M.-l. Flottes, G. Di Natale, B. Rouzeyre, P. Prinetto, and M. Restifo, "Scan chain encryption for the test, diagnosis and debug of secure circuits," in *2017 22nd IEEE European Test Symposium (ETS)*. IEEE, 2017, pp. 1–6.

[19] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure scan and test using obfuscation throughout supply chain," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 9, pp. 1867–1880, 2017.

[20] Y. Atobe, Y. Shi, M. Yanagisawa, and N. Togawa, "Dynamically changeable secure scan architecture against scan-based side channel attack," in *2012 International SoC Design Conference (ISOCC)*. IEEE, 2012, pp. 155–158.

[21] L. Azriel, R. Ginosar, and A. Mendelson, "Exploiting the scan side channel for reverse engineering of a vlsi device," *Technion, Israel Institute of Technology, Tech. Rep. CCIT Report*, vol. 897, 2016.

[22] K. Z. Azar, H. M. Kamali, H. Homayoun, and A. Sasan, "Smt attack: Next generation attack on obfuscated circuits with capabilities and performance beyond the sat attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 97–122, 2019.

[23] L. Alrahis, M. Yasin, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "Scansat: unlocking obfuscated scan chains," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. ACM, 2019, pp. 352–357.

[24] M. El Massad, S. Garg, and M. Tripunitara, "Reverse engineering camouflaged sequential circuits without scan access," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017, pp. 33–40.

[25] E. M. Clarke, W. Klieber, M. Nováček, and P. Zuliani, "Model checking and the state explosion problem," in *Tools for Practical Software Verification*. Springer, 2012, pp. 1–30.