AFRL-RY-WP-TR-2022-0274

HARDWARE INTELLECTUAL PROPERTY (IP) PROTECTION THROUGH PROVABLY SECURE STATE-SPACE OBFUSCATION SECURITY ANALYSIS AND INDUSTRIAL IMPLEMENTATION OF DYNAMICALLY OBFUSCATED SCAN CHAIN (DOSC) ARCHITECTURE PROJECT

Mark Tehranipoor
University of Florida

MARCH 2023
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with The Under Secretary of Defense memorandum dated 24 May 2010 and AFRL/DSO policy clarification email dated 13 January 2020. This report is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RY-WP-TR-2022-0274 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signature//

POMPEI L. ORLANDO
Program Manager
Trusted Electronics Branch
Aerospace Components & Subsystems Division

//Signature//

SKYLER R. HILBURN, Lt Col Chief
Trusted Electronics Branch
Aerospace Components & Subsystems Division

//Signature//

GENE M. WILKINS, Lt Col, USAF
Deputy Chief, Aerospace Components &
Subsystems Technology Division
Sensors Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
March 2023	Final	21 February 2018	8 March 2022

4. TITLE AND SUBTITLE

HARDWARE INTELLECTUAL PROPERTY (IP) PROTECTION THROUGH PROVABLY SECURE STATE-SPACE OBFUSCATION SECURITY ANALYSIS AND INDUSTRIAL IMPLEMENTATION OF DYNAMICALLY OBFUSCATED SCAN CHAIN (DOSC) ARCHITECTURE PROJECT

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8650-18-1-7821	N/A	N/A
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
N/A	N/A	Y1RX

6. AUTHOR(S)

Mark Tehranipoor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
University of Florida 968 Center Dr., Larsen Hall 336A, Gainesville, FL 32611	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command, United States Air Forces	Defense Advanced Research Projects Agency (DARPA/MTO) 675 North Randolph Street Arlington, VA 22203	AFRL/RYDT	AFRL-RY-WP-TR-2022-0274

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This work was funded in whole or in part by Department of the Air Force contract FA8650-18-1-7821. The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U. S. Government. This material is based on research sponsored by the Air Force Research Lab (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-1-7821. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Labs (AFRL), the Defense Advanced Research Projects Agency (DARPA) or the U.S. Government. Report contains color.

14. ABSTRACT

This research investigated the use of logic locking and dynamically obfuscated scan chain (DOSC) to resist IP piracy when manufactured in a commercial cutting-edge semiconductor facility. Logic Locking, also known as Logic Obfuscation, has emerged as a promising solution to resist IP piracy. Logic locking introduces additional gates controlled by key input to conceal original functionality. The correct operation of the design is ensured once the correct unlocking key inputs are provided from a tamper-proof memory. Over the past ten years, researchers have proposed several logic locking methods. However, all these logic locking methods turned out to be breakable. A major attack driving force behind the vulnerability of these logic locking methods came out from the Boolean satisfiability-based (SAT) attacks. There are logic locking techniques that claim to be highly resistant to SAT attacks. Still, their outputs are highly corruptible, and their structural traces are more vulnerable to other attacks such as bypass attacks, signal probability skew attacks (SPS), and removal attacks. Dynamically obfuscated scan chain (DOSC) is complimentary to logic locking and prevents exfiltration of scan chain information used within a SAT attack.

15. SUBJECT TERMS

microelectronics, microelectronics security, logic locking, IP piracy

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
Unclassified	Unclassified	Unclassified	SAR	30

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Pompei Orlando	N/A

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)

Prescribed by ANSI Std. Z39.18

Table of Contents

Section **Page**

List of Figures

Figure **Page**

List of Tables

Table **Page**

1 INTRODUCTION

1.1 Background

The costs associated with maintaining a cutting-edge semiconductor fabrication facility have increased in recent years, which has resulted in the emergence of fabless semiconductor companies, third-party design houses, and contract foundries. Since many companies involved in design, manufacturing, integration, and distribution are in various parts of the world, the original intellectual property (IP) owners can no longer strictly monitor and control the entire process. From a global perspective, where IP laws (and the degree of enforcement) vary significantly from country to country, IP protection and assurance cannot be achieved by passive methods such as patents, copyrights, IC metering, and watermarks that merely deter these threats.

Logic Locking, also known as Logic Obfuscation, has emerged as a promising solution to resist IP piracy. Logic locking introduces additional gates controlled by key input to conceal original functionality. The correct operation of the design is ensured once the correct unlocking key inputs are provided from a tamper-proof memory. Over the past ten years, researchers have proposed several logic locking methods. However, all these logic locking methods turned out to be breakable. A major attack driving force behind the vulnerability of these logic locking methods came out from the Boolean satisfiability-based (SAT) attacks. There are logic locking techniques that claim to be highly resistant to SAT attacks. Still, their outputs are highly corruptible, and their structural traces are more vulnerable to other attacks such as bypass attacks, signal probability skew attacks (SPS), and removal attacks. Figure 1 outlines a chronology of events in logic locking.

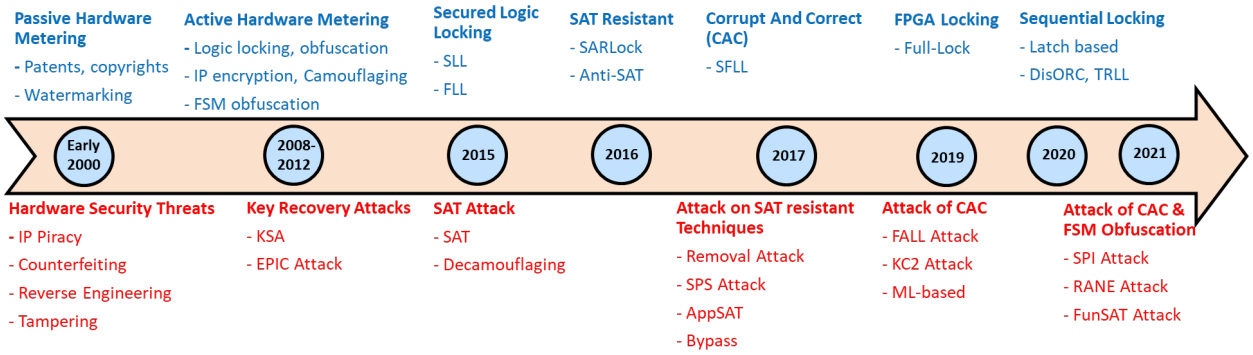

Figure 1: Timeline of Attack and Defense Events in the Last Decade On logic Locking

SAT attack extracts the secret unlocking key by accessing the scan chain to achieve maximum controllability and observability in the unlocked chip. It is clear of Figure 1 that, none of the proposed countermeasures can provide the required resiliency by obfuscating only in functional mode. In summary, the vulnerability of logic locking is prominent from the timeline of events and summarized below.

- **Combinational locking** – tradeoff between corruptibility and SAT resistance, functional/structural trace
- **Point function-based locking** – protects against SAT but leaves backdoor open to bypass/removal attacks

- **Corrupt and Correct (CAC)** – modifying the prime implicant table is fundamentally vulnerable to sparse prime implicant (SPI) attack, hard-coding the key to restoring unit is functionally vulnerable to FALL attack
- **FSM obfuscation** – vulnerable to model-checking based RANE attack, topological RE of state elements, keyless obfuscation methods are vulnerable to FunSAT attack

Hence, in this project, we propose dynamic obfuscation of scan chains to exponentially increase the complexity of performing SAT attacks in a black-box scenario. Even though the attacker, having access to the scan chain, will be getting some information from the unlocked chip, with DOSC in place, that information will act as noise or jargon to the attacker.

1.2 Objective

We adopt one of our prior schemes that were aimed to secure a design against scan-based side-channel attacks by dynamically obfuscating scan chains [26]. We take inspiration from this scan chain protection technique to develop a dynamically obfuscated scan chain (DOSC) for the protection of obfuscated circuits and perform a comprehensive security assessment. Our objectives are as follows.

1. **Comprehensive Security Assessment**
    - Comprehensive mathematical model that covers all security parameters of the DOSC
    - Extend this analysis across different SAT solvers.
2. **Tamper Resistance**
    - Evaluate DOSC effectiveness against different oracle-less and tampering attacks.
    - Implement a self-testability mechanism for DOSC.
    - Malicious circuit detection through observability and controllability analysis
3. **Design, Implementation, Validation**
    - Develop a CAD based solution for DOSC integration.
    - Establishing Secure Test Mechanism using DOSC

We implement an obfuscation tool flow that takes DOSC design parameters as input and automatically produces the DOSC design with minimal area and test time overhead.

1.3 High-level Accomplishments

Here we are highlighting some of the key accomplishments in this project.

- **First-ever scan obfuscation method:**
  In this project, we developed dynamic obfuscation of scan chain, the first-ever technique that protects scan architecture using locking gates. DOSC scrambles the scan chain contents based on pseudo random numbers to deter attackers' capability to carry out a meaningful attack.
- **Unbreakable logic obfuscation method:**
  Since its inception in 2017, DOSC has remained unbreakable despite several attempts made during hardware de-obfuscation competition.

(https://trust-hub.org/#/competitions/hwobfuscation1)

- **Formal characterization of the attack complexity:**
  DOSC is the only solution that provides mathematical proof of complexity like AES,

$$t = O(2^{d.n})$$

Where, t = time to break DOSC, n = number of variables, and d = depends on the density of the SAT instance of DOSC. Here, n, d can be estimated from the mathematical proof.

- **Security Characterization of DOSC architecture:**
  Security DOSC increases the resiliency of functional logic locking exponentially. The resiliency offered by DOSC acts as a layer of a shield on top of the underlying functional logic locking method. Therefore, even if the underlying functional logic locking method is susceptible to SAT-based adversarial attacks, DOSC can protect the functional secrecy.
- **Achieving testability conforming industry practice:**
  DOSC is the only available scan obfuscation method that ensures manufacturing testability. There are a couple of scan obfuscation methods available. However, none of those methods consider how manufacturing testability will be assured while performing scan scrambling. This project also developed a test pattern and response transformation framework that takes DOSC configuration parameters and ATPG patterns as inputs and generates the obfuscated patterns as an output to ensure maximum test coverage.
- **Security analysis against emerging attack:**
  DOSC protects the functional logic-locked design against emerging model-checking-based attacks. Since the acceptance of the project, several emerging attacks have been proposed against logic locking. DOSC provides resiliency even against these emerging attacks that exploit different solvers.
- **Thwarting tampering attacks and trojan insertion:**
  To investigate the resiliency of the DOSC-inserted design against tampering and trojan insertion, we provided three different solutions based on the activity and objective of the trojan. DOSC circuitry is 100% testable, making any confidentiality or integrity violation-based trojan within DOSC easily detectable. For denial of service-based trojans, we proposed an in-house developed technique called ‘trojan scanner’, which can detect any trojans within hours.
- **Deploying a plug-and-play CAD tool in an industrial platform:**
  This project developed a python-based plug-and-play CAD tool that takes the target design netlist as input and inserts DOSC circuitry in the design based on the user-defined key size and other security parameters. The developed CAD tool is design, locking method, key size, and scan architecture agnostic.
- **First-ever Logic locking based ATPG without sharing the secret unlocking key:**
  To achieve adequate test coverage in a logic locking chip, the manufacturing test patterns used to be generated by applying the unlocking key. Therefore, the manufacturing test needed to be performed in a trusted facility or in an untrusted facility with proper key provision mechanisms. Based on the suggestion of the program manager, we developed the first-ever logic-locking aware test pattern generation method that provides a set of dummy keys that maximizes the manufacturing test coverage with minimal test cost overhead.

1.4 Dynamically Obfuscated Scan Chain (DOSC)

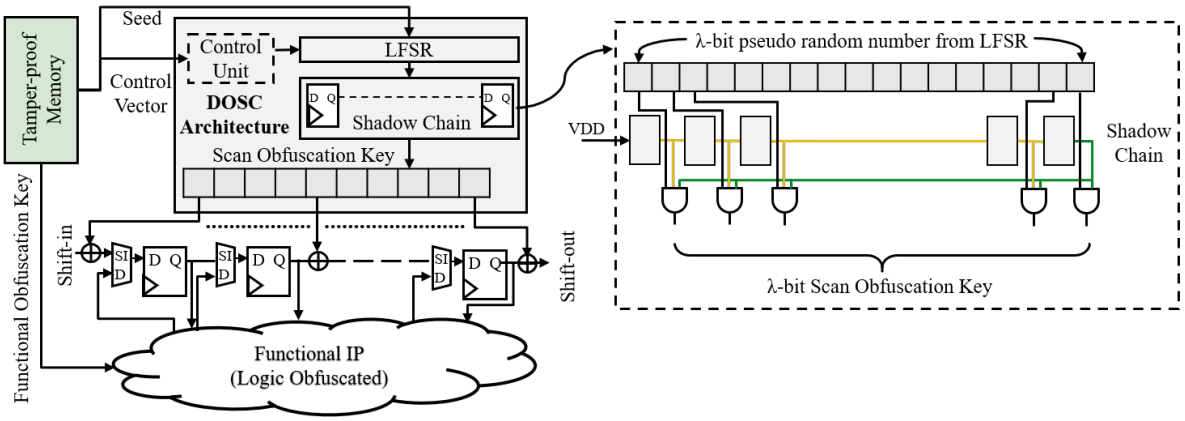

Figure 2: DOSC Architecture and its Different Components

A high-level overview of the DOSC architecture, inserted in the scan chain of a logic-obfuscated functional IP, is shown in Fig. 2. The logic obfuscation of functional IP can be done by existing logic obfuscation schemes. However, the main security advantage comes from the DOSC architecture itself which obfuscates the values extracted from scan chains. The DOSC architecture [29] is composed of three major components:

1) Linear feedback shift register (LFSR): In our proposed DOSC architecture, the scan obfuscation key changes randomly. A polynomial primitive LFSR is adopted to generate λ-bit pseudo-random numbers which are later passed through the shadow chain (to be discussed next) and connected to the scan chain by XOR gates to one-time pad (OTP) scan chain contents. The LFSR reads control signals from the control unit to generate pseudo-random permutation rate, $\alpha = \frac{Scan\ clock\ frequency}{LFSR\ clock\ frequency}$. For example, a permutation rate of $\alpha = 4$ means that LFSR generates pseudo-random numbers once in every 4 scan clock cycles.
2) Shadow Chain: The shadow chain protects LFSR generated pseudo-random scan obfuscation keys from ScanSAT [25] attack and other scan-based attacks. The length of the shadow chain is the same as the LFSR. It takes the λ-bit pseudo-random number generated by the LFSR as input and generates a λ-bit scan obfuscation key. Shadow chain consists of a chain of flip-flops, driven by the scan clock, with the first flip-flop input connected to logic '1'. The outputs of the flip-flops are connected to the XOR gates inserted into the scan chain through a series of AND gates (details on how shadow chain works are discussed in [26]). Upon reset or when a new seed is loaded, at first, all the flip-flops in the shadow chain are reset and forced to logic '1' serially with scan clock frequency. When the last flip-flop of the shadow chain becomes '1' at the λ^{th} clock cycle after reset or seed being loaded, only then is the scan obfuscation key applied to the XOR gates and scrambled responses start showing at the scan-out port.
3) Obfuscated Scan Chain: Obfuscated scan chain is the scan chain of the (logic locked) functional IP with λ number of XOR gates uniformly placed throughout the chain. One of the inputs of the XOR gates comes from the λ-bit scan obfuscation key while the other input comes

from the scan chain. When the scan obfuscation key is applied to the scan chain, the XOR gates OTP scan chain contents.

2 TECHNICAL ACCOMPLISHMENTS

2.1 Summary of Technical Accomplishments

In section 1.2, we discussed about the objectives and tasks of this DARPA hard project. We granulated each of the different tasks into multiple sub-tasks to tackle the problem from a holistic point-of-view. In Table 1, we summarize these subtasks and their associated accomplishments. A bold accomplishment represents a novel work here.

Table 1. Summary of Technical Accomplishments

Tasks	ID	Subtasks	Accomplishment (bold = novel)
Comprehensive Security Assessment	1.1	Comprehensive DOSC model with Security Parameters	**Developed a mathematical model that takes DOSC parameters as inputs and provided quantifiable resiliency**
	1.2	SAT Attack complexity Analysis against Dynamic obfuscation	Provided 3-CNF SAT based Analogy. Reviewed literatus from last 30 years and provided range of exp. complexity
	1.3	Time complexity estimation for oracle-guided attacks	**Provided SAT instance density-based analogy. Performed exhaustive study of existing literatures and provided proof of resiliency.**
	1.4	SAT attack on DOSC using different SAT solvers	**Performed emerging attacks on DOSC. Achieved timeout margin (7 days) with just 24-bit DOSC.**
Tamper resistance & Oracle-less Attack	2.1	Investigation on Oracle-less Attacks	DOSC shows resistance to ML based attacks (SAIL, SWEEP). ML-attacks fail to properly apply the weights of dynamic scan obfuscation in the case of DOSC.
	2.2	Self-testability of DOSC	Demonstrated self-testability feature of DOSC with the existing structures
	2.3	Trojan detection in DOSC Circuitry	Developed exhaustive threat model based on the trojan payload and objective. Provided detection solution for all the different scenarios with inherent property of DOSC
Implementation, design, and validation	3.1	Automated CAD tool development for DOSC	Developed a python-based framework for DOSC that takes design netlist, and scandef as inputs and generates DOSC inserted netlist and scandef as output.
	3.2	Test Pattern and Response Pre-Obfuscation	**Developed pattern and response pre-obfuscation method to achieve maximum manufacturing testability**
	3.3	Valet Key-based ATPG	**Developed LL-ATPG method for dummy key based ATPG that can achieve higher test coverage than the traditional ATPG method**
	3.4	Establishing Secure Test Mechanism using DOSC	**Developed framework to transform ATPG generated patterns for DOSC scenario considering the permutation rate and XOR gate locations. DOSC matches the original test coverage of the design to 100%**

2.2 Additional Accomplishments Outside of the Initial Proposal

Over the course of this 3-year long project, we came across several sub-tasks that needed to be investigated to perform a comprehensive assessment of the related tasks. Therefore, we performed some additional tasks that are summarized in Table 2. These tasks are byproducts of the other associated tasks or sub-tasks.

Table 2. Summary of Additional Accomplishments

Additional Tasks	**Accomplishments**
SAT attack on DOSC using different SAT solvers	Performed any attacks proposed since the acceptance of proposal, e.g., RANE, FunSAT
Trojan detection in DOSC Circuitry	Developed a holistic threat model and provided several approaches for trojan detection based on the objective (CIA) of the trojan
Test Pattern and Response Pre-Obfuscation	Developed pattern and response pre-obfuscation method for both plain and compressed scan chain. Performed exhaustive investigation of different ATPG tools (Tessent EDT, Modus ATPG, TetraMAX)
Valet Key-based ATPG	**Developed LL-ATPG as a plug and play solution for test pattern generation of logic locked chip in general** **Developed method is logic locking type and scan architecture agnostic** **Developed a CAD tool of the LL-ATPG**
Establishing Secure Test Mechanism using DOSC	Developed framework to transform test patterns which is applicable to any dynamic or static scan obfuscation method The framework in logic agnostic method
Functional Locking for DOSC	**Developed FSM based obfuscation method mounted with DOSC architecture by re-suing the existing structures of DOSC** **Performed emerging attacks to prove resiliency of the prototype**

3 TASK AND ACCOMPLISHMENT DETAILS

3.1 Comprehensive Security Assessment

This task comprises the security assessments that are governed by satisfiability algorithm and its variants, providing mathematical model, attack using different solvers etc. Here is a summary of accomplishments under this task category.

- Developed framework to perform different variants of SAT attacks
- Performed the emerging attacks on DOSC, e.g., BMC, FunSAT attack
- Performed SAT attack using different genre of solvers
- DOSC shows exponential resiliency against these attacks
- Performed ML-based attacks against DOSC

3.1.1 Comprehensive DOSC model with Security Parameters

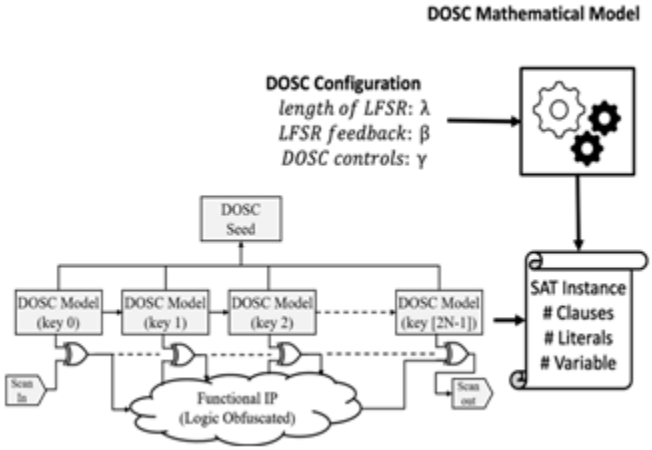

Figure 3: Attack Mode in DOSC-inserted Design

Attack against DOSC-inserted design could be against DOSC seed or functional obfuscation key along with DOSC seed. In this attack method, the attacker models DOSC for dynamic scan obfuscation key of each cycle and performs the SAT attack to trace back to LFSR seed and the functional locking key. From an attacker's perspective, this attack model should be the most promising one to compromise the security of DOSC-integrated functional obfuscated circuit since the seed is the only static asset in DOSC. Furthermore, this attack exploits the minimum complexity bound in a DOSC-inserted obfuscated circuit as it can be performed utilizing test mode only; the functional circuitry is bypassed (represented by the dotted line shown in Fig. 2 so it doesn't introduce any additional complexity. With the knowledge of the seed and the configuration of the LFSR, the attacker can attempt to identify scan obfuscation key at any cycle performing the scrambling translation on its own. This breaks the scan obfuscation security and clears the attacker's path to perform SAT attack to find the functional obfuscation key. We modeled different sizes of benchmarks from ITC'99 and ISCAS'89 circuits using this model presented in Fig. 3 and then performed SAT attack. Attack results are show in Fig. 4. We inserted different size DOSC in the scan chain of three different size benchmarks - small (s838), medium (b07) and, large (s38417). For functional obfuscation, we have followed random logic

locking approach and inserted 32-bit functional obfuscation key gates. For each of the benchmarks, we have performed the SAT attack to reveal both DOSC seed and obfuscation key of functional IP. We have also estimated the complexity of the DOSC-inserted functional obfuscated benchmarks using our mathematical model. SAT attack complexity increases exponentially with increasing DOSC key size for all three benchmarks and out mathematical estimation of complexity is always lower than the actual one. We have considered a timeout margin of 10 days, 20 days, and 30 days respectively, for small, medium, and large benchmark.

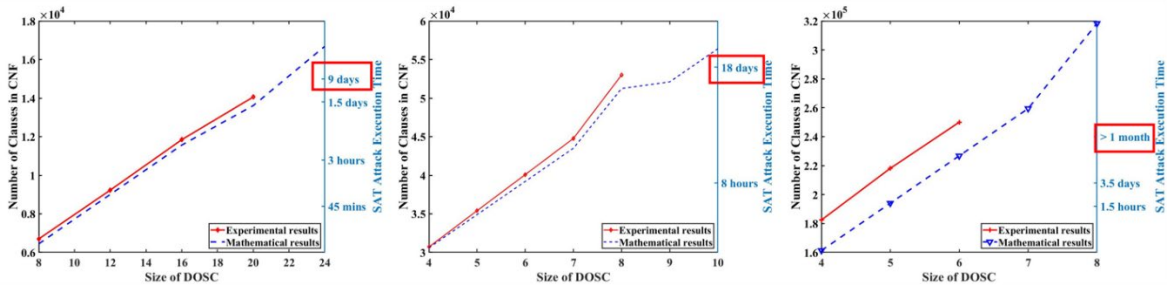

Figure 4: SAT Attack on s838, b14, and s38417 Benchmarks Integrated with Different Size DOSC

3.1.2 Time complexity estimation for oracle-guided attacks

Formulating the complexity of a SAT problem is a fundamental open question. However, researchers have found that the computational complexity of a SAT problem depends on the density (d) of the combinatorial problem, the order of the problem (V), machine resources, the solver type, and solver heuristics. Increasing the size of DOSC increases density ($d = \frac{\#\ clauses}{\#\ variables}$) of SAT instance of the DOSC circuit and always remain in exponential region of complexity. CUDD based solvers (effective for HW verification) faces exponential complexity when density of the SAT instance, $d > 2$. In Figure 5, we have shown how density d of DOSC-inserted design varies with changing DOSC size. From the above explanation and Figure 5, the complexity of the SAT attack on DOSC-inserted design always falls in the exponential region of running time. Therefore, SAT attack execution time of DOSC is mathematically proved to be exponential, $\mathcal{O}(2^{V\varepsilon})$. The designer should choose DOSC architecture parameters, e.g., size of DOSC size, permutation rate, and XOR gate locations such that the attack complexity always grows exponentially. However, the actual time depends on machine resources and solver heuristics.

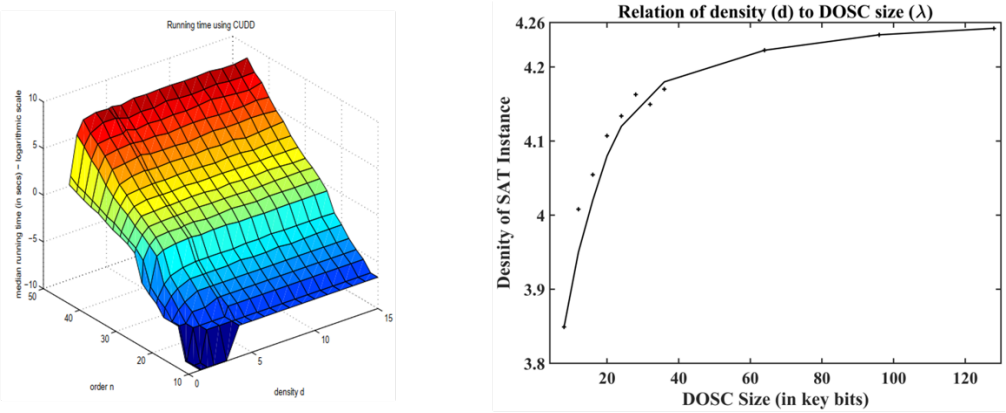

Figure 5: Relation of the Density of DOSC-inserted Benchmark's SAT Instance to DOSC Key Size

3.1.3 SAT Attack on DOSC using Different SAT Solvers

We performed emerging BMC attack [17] on DOSC inserted benchmarks. We assumed that attacker has access to both primary and scan ports for better controllability and observability. There is no unrolling/pre-processing involved in the attack process. We included necessary DFT infrastructure in the attack tool. Timeout margin of 10 days for is considered for the attacks. For b14, the attack time-out for with just a 32-bit size DOSC.

Benchmark	# of Gates	# of FFs	Size of DOSC	Size of RLL	# of Iterations	DIPS Length	Attack Time
b14	~5000	245	6-bit	16-bit	15	8	27 min
			16-bit	16-bit	26	18	4.5 days
			32-bit	16-bit	n/a	n/a	Timeout
s38417	~10000	1636	32-bit	16-bit	n/a	n/a	Timeout

3.1.4 Emerging Attacks on DOSC

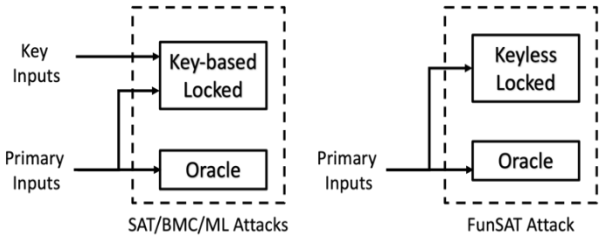

Figure 6: FunSAT vs SAT/BMC/ML Attack Model

FunSAT [18] performs a functional corruptibility (FC) analysis to identify the number of unrolling required to model the sequential flavor. Once the unrolling depth is identified, it performs the regular SAT attack. Primarily, the attack is proposed only on key-less state space obfuscation methods [18]. The attack tool doesn't accept locking benchmarks that require key-inputs [19]. Therefore, the existing FunSAT attack tool is not applicable on DOSC (confirmed by authors).

3.1.5 Oracle-less Machine Learning Based Attack on DOSC

SWEEP attack [20] is an oracle-less machine learning based attack on logic locking that tries to predict the synthesis optimization rules and then revert to the original netlist prior locking. To attack a benchmark locked with RLL, SWEEP will need several benchmarks (does not have to be the same as attacked benchmark) locked with exactly same locking method. Thus, to guarantee a successful attack, attackers need to have the knowledge of the benchmark-locking implementations. For different logic locking methods, the weights need to be revisited. Brings low scalability concern of the SWEEP attack, because even same locking method would have different implementations, which cause differences on the trained feature weights. E.g., if RLL is implemented with XOR rather than MUX (as in the paper) the weights need adjustments. Contacted the author for this concern regarding weight adjustment.

3.2 Tamper Resistance and Trojan Detection

The objective of this task is to investigate all the non-satisfiability attacks that does not take advantage of oracle and any tampering in DOSC inserted design. Here are the key accomplishments achieved under this task –
Investigated the trojan objective, types exhaustively and provided solution for each scenario
DOSC can detect any trojan with confidentiality and integrity violation with 100% accuracy

3.2.1 Threat Model

For trojan detection and tamper resistance, we considered an untrusted foundry threat model. Different components of the threat model are discussed in the table below.

Model Attributes	
Asset	DOSC seed, LFSR states, DOSC flow
Attacker capability	Access to the GDS, and netlist
Attack surface	• Debug ports • Primary ports
Attack technique	• Leak the asset through attack surface • Bypass the secure flow • Bypass scan obfuscation

3.2.2 Trojan Analysis and Detection Mechanism within DOSC

Following table shows different types of trojan based on the payload and how it can be detected within DOSC circuitry.

Types of Trojan	Objective	Example	Detection Approach
Confidentiality	Leaks confidential data covertly to the adversary	Extract the secret seed	Can be detected by DOSC self-testability
Integrity	Allows unauthorized access to a privilege system	Extract the LFSR states	Can be detected by DOSC self-testability
Denial-of-service	Causes system functionality to be unavailable when needed	Bypass scan obfuscation	Use trojan scanner [1]

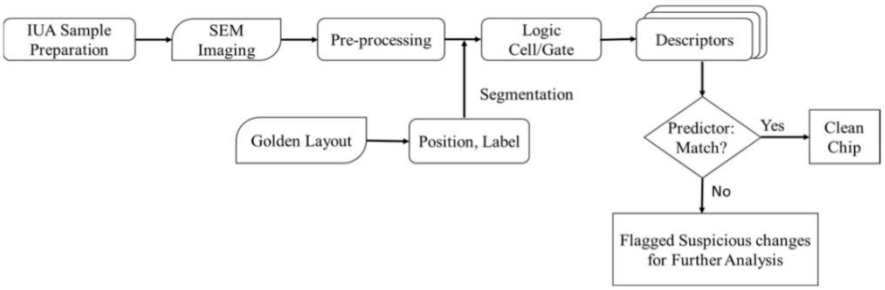

Figure 7: Trojan Detection Framework in Trojan Scanner [16]

```
    Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                     code   #faults
------------------------------  ----  ---------
Detected                         DT      12536
Possibly detected                PT          0
Undetectable                     UD          4
ATPG untestable                  AU          0
Not detected                     ND          0
-----------------------------------------------
total faults                             12540
test coverage                          100.00%
-----------------------------------------------
          Pattern Summary Report
-----------------------------------------------
#internal patterns                          14
    #basic_scan patterns                    14
-----------------------------------------------
          CPU Usage Summary Report
-----------------------------------------------
Total CPU time                           0.21
-----------------------------------------------
```

Figure 8: Trojan Detection Coverage in DOSC Circuitry

In case of stealthy trojans that does not create any confidentiality or integrity violation, simulation-based detection would not be possible. Full chip reverse-engineering is the one possible approach which takes months. Trojan scanner can detect such trojans within hours by comparing the SEM imaging of the fabricated chip with the layout of the golden GDS.

DOSC self-testability:
DOSC architecture has a 100% test coverage even for a size of 256-bit. Therefore, any confidentiality or integrity-based violation within DOSC, should be detected. We can also utilize trojan scanner for such detection, but that would take hours compared to seconds in this method. Please, note that trojan scanner can also be used to detection any trojan with confidentiality or integrity violation. However, trojan scanner will take much longer time than ATPG based self-testability method.

3.3 Implementation, design, and testability

The objective of this task is to develop a computer-aided design (CAD) tool for DOSC generation, implementation, and testability. Some key accomplishments in the task are highlighted below.
Accomplishments:

- Developed python-based plug and play tool that takes design inputs, DOSC configurations params and generates DOSC-inserted netlist, scandef, stil file in seconds
- Integrated DOSC with CEP up to physical layout. DOSC achieves < 1% PPA overhead even with 256-bit key
- Developed method and CAD tool for dummy key-based test generation irrespective of locking type & scan
- Developed test pattern and response offline obfuscation method for plain and compression scan chain

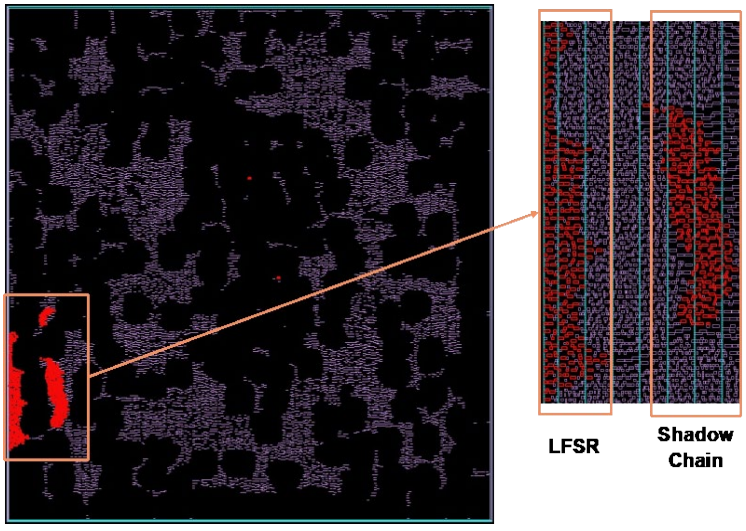

Figure 9: GPS Core with 64-bit DOSC (Synopsys 32nm)

3.3.1 Power, Performance, and Area (PPA) Overhead Analysis in CEP

To investigate the PPA overhead of DOSC inserted design, we have choses GPS Core and DSP core from the CEP SoC. The functionality of this module is not public and adversaries' possess full scan access. Negligible area and power overhead. No impact on performance. Mostly a public knowledge however their efficiency depends on their filter coefficients. Therefore, to hide the coefficients, DSP blocks should be scan locked. We implemented 64-bit DOSC and inserted into the scan chain of GPS core and DSP core. Different attributes of the baseline and DOSC-inserted design's area, power, and performance overhead are shown in the table below. From the above table, it can be observed that, the PPA overhead is < 1% even for smaller designs like GPS/DSP core.

Attributes	GPS Core		DSP Core (Filters)	
	Baseline	DOSC Overhead	Baseline	DOSC Overhead
# Combinational Gates	**10018**	**1.26%**	85965	**0.6%**
# Sequential Gates	126763	0.3%	80732	0.2%
# of Total Gates	136877	0.3%	165577	0.37%
Total Area (um^2)	395996	0.6%	23428.8	0.65%
Leakage Power (mW)	2.53	0.8%	3.21	0.7%
Timing delay (ns)	+18.38	0%	+15.2	0%

3.3.2 Test Pattern Transformation Framework for DOSC-inserted Design

A step-by-step test flow in a DOSC facilitated design is shown in Fig. 10 along with a traditional test flow.

As shown in Fig. 10, test engineers must apply obfuscated test patterns. Knowing scan obfuscation keys for each test cycle and the XOR gates' location throughout the scan chain, test engineers can generate obfuscated test patterns. While the test engineer shifts in these obfuscated patterns, DOSC transforms them into original ATPG patterns before switching to functional mode.

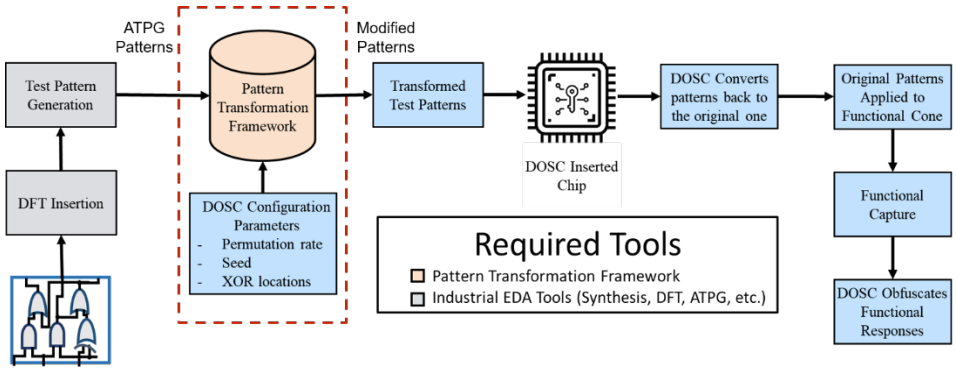

Figure 10: DOSC Facilitated Secure Test Flow

Correspondingly, when the test engineer shifts-out captured functional response, DOSC obfuscates shift-out responses, as shown in Fig. 10. The test engineer receives obfuscated ATPG responses from the designer and compares them with in-field responses to identify defective parts.

In the case of designs with scan compression, compressed test patterns are first decompressed using the decompressor transfer function and then transformed into obfuscated patterns. Similarly, captured responses are compressed first by utilizing the compressor transfer function before obfuscating them using scan obfuscation keys generated by DOSC.

The ATPG pattern and response conversion can be done offline at any trusted facility. The test pattern and response conversion can only be done correctly if the LFSR seed is known, along with the exact architecture and XOR placement of DOSC. Therefore, an adversary who has no access to the seed stored in a tamper-proof memory and is trying to obtain the seed and functional obfuscation keys will have no means to perform such conversions and perform the SAT attack.

A secure test flow considering test compression, their associated pattern, and response transformation is a part of our on-going research.

3.3.3 Secure Test Flow with Compression

In this subsection we explain how pattern transformation takes place in case of a design with combinational test compression where decompressors are built with multiplexers and XOR gates. There are several scan obfuscation methods available in the community. None of these methods discuss how manufacturing testability is going to be taken care of, especially in the case of test compression. Figure 11 shows how decompressors are design during DFT synthesis. In the compressed scan inputs, there are two categories of inputs – i) load more pins (sel[0] in the Fig. 11), and ii) regular scan-in data pins (din[i] in the Fig. 11). When regular scan-in data pins have mix of 1's and 0's, then we can decide 'sel' value based on the model. When regular scan-in data pins are all equal, e.g., all-zeros or all-ones, its not possible to conclude whether a 'sel'=1 or 0 passed that value. In cases where no conclusion can be made for 'sel' value, we guessed a value. Note that scan data inputs are getting exact values, therefore, a 'sel'=0/1 selected it, should not be an issue for testing. Based on this analysis, we developed a pattern transformation framework shown Fig. 12 by utilizing the modeling of decompressor and compressor. Fig. 12 provides the overview of the framework. The framework is up and running. It takes Scan-in and Scan-out patterns from original patterns in STIL file as input. After the transformation, the tool generates the "Modified Scan-in" and "Modified Scan-out" based on our designed 'inverse' compressor/decompressor and functional model of DOSC. The transformation tool shares the modified STIL file with the foundry or OSAT instead of the ATPG generated STIL file. To accomplish this task, we have modelled CoDec for variable debug ports and variable scan chains. Variable scan in/out vs. modelled scan chains for them are shown in the table below:

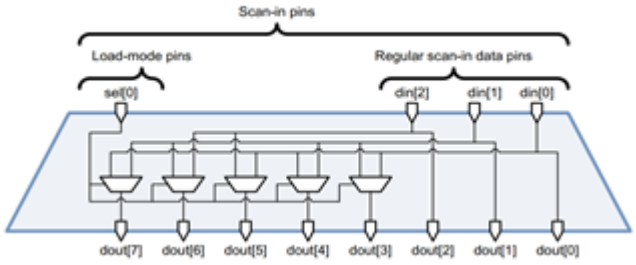

Figure 11: Decompression Structure in Synopsys

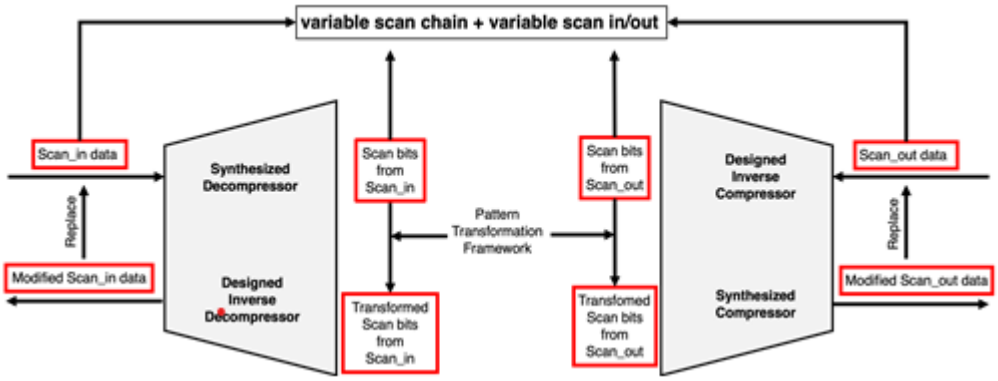

Figure 12: Example Workflow of Automatic test Pattern Transformation

Number of Scan In/Out Ports	Number of Parallel Scan Chains
2	3-10
3	4-14
4	5-16

Additional features of the framework:

1. The tool is developed in MATLAB that takes the design name, and STIL file as input and generates the pre-obfuscated patterns.
2. The tool is generic for a wide range of scan configuration including different compression ratio, number of debug ports, etc.
3. The tool has been tested so far on a number of ITC'99 benchmarks and CEP v3.41 cores.

develop a 3rd party flow for tessent EDT tools.

In the following figure 13, we are presenting a simulation result of the framework from Fig. 12. Here in Fig. 13(a), we performed fault simulation of the IP core in the Synopsys TetraMAX tool using the original patterns. In the Fig. 13(b), we performed fault simulation of the IP core using the transformed patterns where the test data is 15% different due to the randomness coming from modeling of the load mode pins. However, it can be observed that, despite the different in test data, the fault coverage is exactly same.

Simulation performed for 457799 faults on circuit size of 132625 gates.

#patterns simulated	#faults detect/active	test coverage	process CPU time
32	332051 125748	74.90%	0.34
64	54542 71206	85.59%	0.42
96	23193 48013	90.14%	0.47
128	12221 35792	92.53%	0.51
160	7825 27967	94.07%	0.54
192	5105 22862	95.07%	0.57
224	3474 19388	95.75%	0.59
256	2689 16699	96.27%	0.61
288	2181 14518	96.70%	0.63
320	1659 12859	97.03%	0.65
352	2697 10162	97.56%	0.67
384	1132 9030	97.78%	0.69
416	821 8209	97.94%	0.71
448	613 7596	98.06%	0.72
480	284 7312	98.11%	0.74
512	78 7234	98.13%	0.75
518	13 7221	98.13%	0.76
523	11 7210	98.13%	0.79
528	10 7200	98.14%	0.82

Fault simulation completed: #patterns=529, CPU time=0.84

(a)

Simulation performed for 457799 faults on circuit size of 132625 gates.

#patterns simulated	#faults detect/active	test coverage	process CPU time
32	332051 125748	74.90%	0.36
64	54542 71206	85.59%	0.44
96	23193 48013	90.14%	0.49
128	12221 35792	92.53%	0.53
160	7825 27967	94.07%	0.57
192	5105 22862	95.07%	0.60
224	3474 19388	95.75%	0.62
256	2689 16699	96.27%	0.64
288	2181 14518	96.70%	0.67
320	1659 12859	97.03%	0.69
352	2697 10162	97.56%	0.70
384	1132 9030	97.78%	0.72
416	821 8209	97.94%	0.74
448	613 7596	98.06%	0.76
480	284 7312	98.11%	0.77
512	78 7234	98.13%	0.79
518	13 7221	98.13%	0.80
523	11 7210	98.13%	0.83
528	10 7200	98.14%	0.85

Fault simulation completed: #patterns=529, CPU time=0.88

(b)

Figure 13: Fault Simulation in the IP Core

(a) the original patterns, and (b) the transformed patterns for DOSC insertion with scan compression

3.3.4 3.3.4 LL-ATPG: Valet Key-based Test Pattern Generation

Goal is to identify suitable valet keys that ensures adequate test coverage with minimal test cost overhead without sharing the secret unlocking key with the OSATs. In this work we are considering functional logic locking for evaluating the LL-ATPG method. We consider both plain scan chain and scan compression for LL-ATPG. The developed test pattern generation technique should be logic-locking method agnostic. We treat key inputs as PIs and apply key constraints as '*valet keys*' that does not activate the chip but can perform manufacturing test. The table below shows an experimental evaluation of LL-ATPT flow. Here we generated test patterns of GPS core from CEP using the iterative nature by increasing the target test coverage. It can be observed from the table that LL-ATPG can achieve a target test coverage of 99.95% with just 8 valet keys.

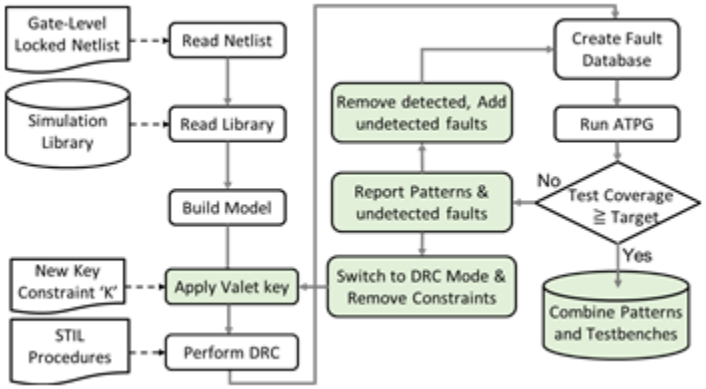

Figure 14: LL-ATPG Flow [21]

Target Coverage	99.5%	99.75%	99.95%	Combined
Actual Coverage	99.65	99.75	99.95	99.95
Total Faults	1093376	3779	2716	1093376
Detected Faults	1089597	1063	2179	1092839
Undetected Faults	3779	2716	537	537
#Valet Keys	2	1	5	8
#Patterns	600	39	254	858

To know further about LL-ATPG and details of the experimental results along with security analysis, we refer the readers to the recent publication [21].

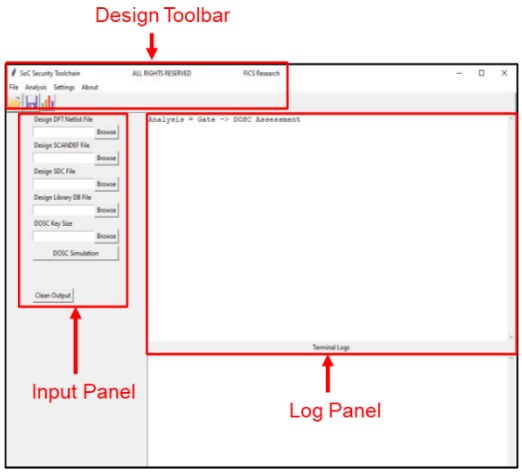

Figure 15: DOSC Insertion Tool

4 COMMERCIALIZATION OF DOSC

DOSC is a patented technology (US20200065456A1). DOSC is currently licensed by Caspia Technologies (https://caspiatechnologies.com/). Part of the main IPPx engine for IP protection in an untrusted foundry environment. A CAD tool framework has been developed with following input/output capability to help designers integrate DOSC with the functional IP core automatically.
Input files – design netlist, scandef, SDC, library, DOSC key size
Output files – DOSC inserted netlist, scandef, SDC files

5 RESEARCH ACCOMPLISHMENTS

Several research papers and patents has already been published in this project and some others are under review.

- Rahman, M.S., Nahiyan, A., Rahman, F., Fazzari, S., Plaks, K., Farahmandi, F., Forte, D. and Tehranipoor, M., 2021. Security Assessment of Dynamically Obfuscated Scan Chain Against Oracle-guided Attacks. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, *26*(4), pp.1-27.
- Rahman, M.S., Li, H., Guo, R., Rahman, F., Farahmandi, F. and Tehranipoor, M., 2021, October. LL-ATPG: Logic-Locking Aware Test Using Valet Keys in an Untrusted Environment. In *2021 IEEE International Test Conference (ITC)* (pp. 180-189). IEEE.
- Tehranipoor, M.M., Forte, D.J., Farahmandi, F., Nahiyan, A., Rahman, F. and Rahman, M.S., University of Florida Research Foundation Inc, 2020. *Protecting Obfuscated Circuits Against Attacks That Utilize Test Infrastructures*. U.S. Patent 16/535,795.

6 CONCLUSION

In this project, we have developed and evaluated, in details, the security of dynamic scan obfuscation (DOSC) scheme that restricts effective scan access to authorized users to protect against oracle-guided attacks and demonstrated both mathematically and experimentally how this architecture can combat SAT attack for extracting logic obfuscation keys. We have performed SAT attack on different DOSC-inserted benchmarks and shown that the time increases exponentially with DOSC key length. We have investigated the resiliency of DOSC against emerging oracle-guided and oracle-less attacks along with different solvers. We developed threat model for tampering and outlined the trojan detection following a holistic approach. We performed PPA analysis in the presence of DOSC in an SoC environment with multi-million gates. As DOSC performs scan obfuscation, we developed framework for manufacturing testability to conform with industry standards. We developed a CAD tool framework for smooth integration of DOSC and automated test data transformation.

7 ACKNOWLEDGEMENT

We would like to thank the program manager of DARPA for their continuous support in the exciting DARPA hard project. We also want acknowledge the constructive commnets from the Government team, MIT-LL, Cadence, and ISI (USC).

8 BIBLIOGRAPHY

[1] Shakya, Bicky, et al. "Introduction to hardware obfuscation: Motivation, methods and evaluation." *Hardware Protection through Obfuscation*. Springer, Cham, 2017. 3-32.
[2] J. A. Roy et al., “Epic: Ending piracy of integrated circuits,” in 2008 ACM DATE, pp. 1069-1074.
[3] J. Rajendran et al., “Fault analysis-based logic encryption,” IEEE Transactions on computers, vol. 64, no. 2,pp. 410-424, 2013.
[4] J. Rajendran et al.,“Security analysis of logic obfuscation,” in 2012 ACM DAC, pp. 83-89.
[5] P. Subramanyan et al., “Evaluating the security of logic encryption algorithms,” in 2015 IEEE HOST. IEEE, 2015,pp. 137–143.
[6] M. Yasin et al. 2016. SARLock: SAT Attack Resistant Logic Locking. In IEEE Int’l Symposium on Hardware Oriented Security and Trust (HOST). 236–241.
[7] Xie, Yang, and Ankur Srivastava. "Anti-SAT: Mitigating SAT attack on logic locking." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38.2 (2018): 199-207.
[8] M. Yasin et al. 2017. Provably Secure Logic Locking: from Theory to Practice. In ACM SIGSAC Conference on CCS. 1601–1618.
[9] X. Xu et al. 2017. Novel Bypass Attack and BDD-based Tradeoff Analysis against All known Logic Locking Attacks. In Int’l Conference on CHES. 189–210.
[10] Yasin, Muhammad, et al. "Security analysis of anti-sat." *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017.
[11] M. Yasin et al. “Removal attacks on logic locking and camouflaging techniques”, IEEE Transactions on Emerging Topics in Computing 8, 2 (2017), 517–532.
[12] Shamsi, Kaveh, et. al., "AppSAT: Approximately deobfuscating integrated circuits." *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2017.
[13] Sirone, Deepak, et. al., "Functional analysis attacks on logic locking." *IEEE Transactions on Information Forensics and Security* 15 (2020): 2514-2527.
[14] Alrahis, Lilas, et al. "ScanSAT: Unlocking obfuscated scan chains." *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. 2019.
[15] Alaql, Abdulrahman, et.al., "Sweep to the secret: A constant propagation attack on logic locking." 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). IEEE, 2019.
[16] Vashistha, Nidish, et al. "Detecting hardware trojans inserted by untrusted foundry using physical inspection and advanced image processing." *HaSS* 2.4 (2018): 333-344.
[17] Roshanisefat, Shervin, et al. "RANE: An Open-Source Formal Deobfuscation Attack for Reverse Engineering of Logic Encrypted Circuits." *Proceedings of the 2021 on Great Lakes Symposium on VLSI*. 2021.
[18] Hu, Yinghua, et al. "Fun-SAT: Functional Corruptibility-Guided SAT-Based Attack on Sequential Logic Encryption." arXiv preprint arXiv:2108.04892 (2021).
[19] https://github.com/descyphy/Fun-SAT
[20] Alaql, Abdulrahman, Domenic Forte, and Swarup Bhunia. "Sweep to the secret: A constant propagation attack on logic locking." *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2019.
[21] Rahman, M. Sazadur, et al. "LL-ATPG: Logic-Locking Aware Test Using Valet Keys in an Untrusted Environment." 2021 IEEE International Test Conference (ITC). IEEE, 2021.

LIST OF ACRONYMS, ABBREVIATIONS, AND SYMBOLS

ACRONYM	DESCRIPTION
CAC	Corrupt and Correct
CAD	Computer-Aided Design
DOSC	Dynamically Obfuscated Scan Chain
FC	Functional Corruptibility
IP	Intellectual Property
LFSR	Linear Feedback Shift Register
OTP	One-Time Pad
SAT	Satisfiability-Based
SPS	Signal Probability Skew