Secure Scan and Test Using Obfuscation Throughout Supply Chain

Xiaoxiao Wang, *Member, IEEE*, Dongrong Zhang, Miao He, *Student Member, IEEE*, Donglin Su, and Mark Tehranipoor, *Senior Member, IEEE*

***Abstract*—Scan-based test is commonly used to increase testability and fault coverage, however, it is also known to be a liability for chip security. Research has shown that intellectual property (IP) or secret keys can be leaked through scan-based attacks, which can be performed by entities within the supply chain. In this paper, we propose a design and test methodology against scan-based attacks throughout the supply chain, which includes a dynamically obfuscated scan (DOS) for protecting IP/integrated circuits (ICs). By perturbing test patterns/responses and protecting the Obfuscation Key, the proposed architecture is proven to be robust against existing noninvasive scan-based attacks, and can protect all scan data from attackers in foundry, assembly, and system development without compromising the testability. Further, a novel test methodology cooperating with the DOS design is also proposed, which shows full pattern application flexibility. Finally, detailed security and experimental analyses have been performed on ITC and industrial benchmarks. Demonstrated by the simulation results, the proposed architecture can be easily plugged into EDA generated scan chains without generating a noticeable impact on conventional IC design, manufacturing, and test flow. The results demonstrate that the proposed methodology can protect chips from existing brute force, differential, and other scan-based attacks that target the Obfuscation Key. Furthermore, the proposed design is of low overhead on area, power consumption, and pattern generation time, and there is no impact on test time.**

***Index Terms*—Obfuscation, scan-based attacks, secure scan, supply chain, testability.**

I. Introduction

SCAN-BASED test is one commonly practiced design-for-test (DFT) scheme due to its high controllability and observability. Jeopardized by the worldwide integrated circuit (IC) supply chain, it can also be used to assist noninvasive attacks, thereby compromising security. The exposed scan chains may leak critical information, such as intellectual property (IP) or secret keys to attackers [1], [2], which can be any entity within the IC supply chain. Hence, practical solutions are needed to protect ICs against scan-based side-channel attacks [3].

In the last decade, there have been a number of scan-based attacks on various crypto systems. In [5], the risk of scan-based attack is presented as a general threat to stream cipher. To obtain critical information, the attackers can ascertain the internal structure of the scan chain by running encryption in normal mode and then switching to test mode. References [6]–[9] have successfully uncovered scan-based attacks on the dedicated hardware implementation of the data encryption standard (DES), elliptic curve cryptosystems, advanced encryption standard (AES), and RSA. Since scan chains directly reveal the internal state of the logic blocks, attackers can use them to perform IP piracy [10]. With the knowledge of the design [11], attackers can also illegally control the chip by scanning in illegal values into the system status registers to disrupt the chip. In light of these threats, ensuring scan security has become a great concern to industry, and various countermeasures have been proposed, including the following.

1) *Defusing the Scan Related Pins:* The most direct solution is to defuse the polysilicon fuses connecting the scan in or scan enable pins [12]; however, this prohibits in-field testing.
2) *Test Mode Protection:* By carefully designing the test controller, test mode request will reset the registers and wrap the nonvolatile memories [13]–[16]. However, a new test mode only attack has been successfully demonstrated in [17].
3) *Advanced Industrial DFT Techniques:* On-chip compression, X-tolerance, and X-masking are considered natural barriers to scan-based attacks [18]. However, the compression bypassing mode is always kept for the sake of debugging and diagnosis. Recently, some attacks have been made even in the presence of on-chip compression [19], X-masking [20], and X-tolerance [21], [22].
4) *Scan Interface Encryption:* In [23], the scan patterns/responses are decrypted/encrypted at each scan input/output, respectively, which is conducted by highly efficient and secure block cipher at each scan port.
5) *Partial Scan:* The secure scan architectures presented in [8], [24], and [25] exclude flip-flops containing sensitive information from the scan chain. However, only part of the scan chain cells can be protected. Besides, defects in the excluded registers cannot be detected,

Manuscript received May 24, 2017; revised August 23, 2017; accepted October 25, 2017. Date of publication November 13, 2017; date of current version August 20, 2018. This work was supported by the National Science Foundation of China under Grant 61631002, Grant 61504007, and Grant 61427803. This paper was recommended by Associate Editor N. Nicolici. *(Corresponding author: Xiaoxiao Wang.)*

X. Wang, D. Zhang, and D. Su are with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: wangxiaoxiao@buaa.edu.cn).

M. He and M. Tehranipoor are with the ECE Department, University of Florida, Gainesville, FL 32608 USA (e-mail: tehranipoor@ece.ufl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2017.2772817

TABLE I
EFFECTIVENESS OF EXISTING COUNTERMEASURES FOR PROTECTING IPS AGAINST SCAN-BASED ATTACKS IN SUPPLY CHAIN

Supply Chain	IC Integrator (Design House)	Foundry	Assembly/Test Facilities	OEM/EMS	Distributor	End Customer
Defusing the Scan Related Pins	×	×	×	✓	✓	✓
Test Mode Protection	×	×	×	○	○	○
Advanced Industrial DFT Technique	○	○	○	○	○	○
Partial Scan	✓	✓	✓	✓	✓	✓
Obfuscated Scan	○	○	○	○	○	○
Scan Chain Reordering	○	○	○	○	○	○
Dynamically-Obfuscated Scan [4]	✓	✓	✓	✓	✓	✓

(1) ✓ indicates that the countermeasure can protect IPs against all scan-based attacks from corresponding role in supply chain;
(2) ○ indicates that the countermeasure can protect IPs against some types of scan-based attacks from corresponding role in supply chain;
(3) × indicates that the protection is ineffective.

which decreases the test coverage potentially impacting yield.

6) *Obfuscated Scan:* In [26]–[33], dummy flip-flops or other obfuscation logics (i.e., inverters, XOR gates, etc.) have been inserted into the scan chain to randomize scan outputs. A scan chain access authorization process usually controls obfuscation. However, some obfuscation logics inserted into the scan chain are not robust against reset or flushing attacks [26], [27]. More importantly, the scan authorization key bits hidden in the test patterns are usually easy to locate [28]–[31].
7) *Scan Chain Reordering:* In [11], the order of scan cells is dynamically reconfigured by an unpredictable scrambler, which increases the routing overhead significantly. In [34], each scan chain is divided into several segments, and then the test controller determines the segments' scanning out sequence. In [5] and [35], scan tree architecture is applied to reorder the scan chains. However, these methods still could not defend against a differential attack [19], and require significant change of DFT flow.

From Table I, it can be seen that most of the existing countermeasures cannot provide full protection for all types of attacks, or can only provide protection for part of the supply chain. For example, countermeasures defusing the scan related pins prevent IC from being attacked by original equipment manufacture (OEM), electronic manufacturing supplier (EMS), distributor, and end customer, however, they expose un-encrypted test patterns and responses to IC integrator, foundry, and assembly. This enables counterfeiting, and IP piracy at early stages of the supply chain. Although partial scan provides protection throughout the supply chain, the extensive use of partial scan can significantly lower test coverage.

To address the above-mentioned security shortcoming, this paper presents a novel dynamically obfuscated scan (DOS) design, as well as a test methodology, which have the following advantages.

1) It can protect IPs against existing noninvasive scan-based attacks while maintaining the testability and pattern application flexibility.
2) It can prevent vital information from being stolen by malicious users throughout the supply chain, i.e., foundry, assembly, and in distribution.
3) It offers low area/power overhead, has little impact on industrial design and test flow, and there is no increase in test time.

The rest of this paper is organized as follows. Section II explains the threat models leading to our protection objectives. Section III describes the proposed architecture. The DOS-based test methodology is presented in Section IV. Section V introduces the design implementation flow considering the DOS design. Section VI provides experimental results. Finally, concluding remarks are given in Section VII.

II. THREAT MODELS AND PROTECTION OBJECTIVES

This section presents the threat model, which sets up the protection objectives that the proposed solution should satisfy.

A. Threat Models

An attacker in the supply chain tries to use the scan chain (sometimes through JTAG [36]) to:

1) steal critical information from crypto IP [5]–[9];
2) violate confidentiality and integrity policies [37];
3) pirate IP design [10], [38];
4) illegally take control of the chip [11].

The scan-based noninvasive attack methods making such malicious acts possible are as follows.

1) *Scan Facilitated Differential Attack:* Differential attack has been proposed in [38] and [39]. By inputting challenge pairs, running the crypto algorithm, and comparing the outputs, the key can be obtained. This attack has been facilitated by scan due to added controllability and observability. Through switching from functional mode to test mode, the attacker can identify key flip-flops from the scan chain. Then the key can be recovered through the already constructed correlation among input pairs, key flip-flops, and key [40]. Although some test mode protection techniques attempt to reset data registers when the chip is switched to test mode, test mode only differential attacks have recently been discussed in [19]. Furthermore, differential attacks are reported even in the presence of advanced DFT structures, i.e., on-chip compression, X-masking, and X-tolerance [19]–[22].
2) *Attacks Designed for Specific Countermeasures:* In addition to the on-chip compression used in DFT structures, scan chain reordering and obfuscation have been developed as countermeasures, which can be defeated by the following attacks.

a) *Resetting and Flushing Attacks:* By resetting the scan cells or flushing the scan chain with the known patterns, the fixed inverted bits [26] and modified bits [27] in the obfuscated scan chain can be identified so that the plain text can be deciphered.

b) *Bit-Role Identification Attack:* For countermeasures using the key & lock scheme [28], [30], [31], [34], [41], the scan out responses are determined by the test authentication status. The authentication key bit flipping would make scan out vectors differ, while a nonkey bit would not. This would significantly reduce the difficulty of identifying the key bits (especially for malicious users in fab or assembly).

3) *Combinational Function Recovery Attack:* Since the scan chains unfold the sequential logic as combinational and directly reveal the internal state of the circuit, extracting design information from them has become easier. Thus, the device's functionality can be reverse engineered [10].

B. Threat Models Applicable to Different Stages of Supply Chain

In supply chain, an IC needs to go through IC integrator (SoC design house), foundry, assembly/test facilities, OEM, EMS, distributor, and end customer [42]. Thus it is worthy to analyze the security risks due to scan-based attacks at each stage.

1) *IC Integrator:* Here, the IC integrator refers to the members belonging to IC design house (or IP owner), who integrates custom logic, 3PIPs, and peripherals macros to form the whole IC. In other words, an IC integrator can be either a design, verification, DFT, or even a firmware engineer within the IC design house. Hence the threat model is that, during integration, the confidentiality and integrity policies can be violated by a malicious IC integrator. For example, a malicious frontend RTL designer, who is eligible to access the RTL code, can leak function of IP cores.

2) *Foundry:* Before wafer slicing, all individual dies are tested on wafer by applying scan-based test patterns and scanning out test responses to automatic test equipment (ATE). Similarly, a malicious foundry can pirate IP design utilizing the unobfuscated full-scan. Furthermore, some of the existing secure scan solutions have been proven to be insecure. Thus, sensitive information (i.e., keys, seeds, etc.) for secure scan can be the target for malicious foundry.

3) *Assembly/Test Facilities:* As described in [43], for many cases, the structural test carried by foundry on wafer is enough for quality assurance. However, for some IC design houses making chip for industrial (i.e., automotive) and military applications, the packaged ICs are required to be thoroughly scan tested after packaging, which extends the scan accessibility to assembly/test and even OEM/EMS. Hence, the risks for scan-based attacks at assembly/test facilities are similar to those at foundry.

4) *OEM/EMS:* At OEM or EMS, the printed circuit board (PCB) or equipment carrying IC is developed. At the same time, the IC is programmed/configured to work with the system. To keep in-filed failure analysis ability, usually scan chains are accessible through JTAG interface [6]. The keys and seeds for crypto IP (such as AES, DES, or RSA) loaded into IC in these stages can be leaked. Moreover, programs illegally utilizing scan chain to control the IC can also be loaded into the device in this stage.

5) *Distributor:* The distributor loads customized programs into IC for different customers as required. The risks in this stage are similar to OEM/EMS.

6) *End Customer:* Malicious end customers (or hackers) may be interested in the sensitive information stored in the IC (i.e., device configuration bits). The scan chain accessibility makes crypto IP (such as AES, DES, or RSA) vulnerable. Scan chains can also be used by attackers to illegally control the IC.

In summary, there are many opportunities for malicious entities within supply chain to exploit the scan chains. Fig. 1 shows the attacks each malicious entity in the supply chain can potentially carry out. Therefore, protecting IPs against scan-based attacks throughout the supply chain is necessary.

C. Protection Objectives and Assumptions

With a decade of development, a number of secure scan techniques have been proposed. However, the drawbacks of existing techniques were summarized in Section I. Although the *general protection objective* is to protect IPs against scan-based attacks throughout supply chain, based on the risk distribution, the *general protection objective* can be separated as follows.

Subobjective 1: The proposed architecture should be able to prevent sensitive information leakage from crypto IP by foundry, OEM, and so on.

Subobjective 2: The proposed architecture should be able to disable confidentiality and integrity policies violation by IC integrator.

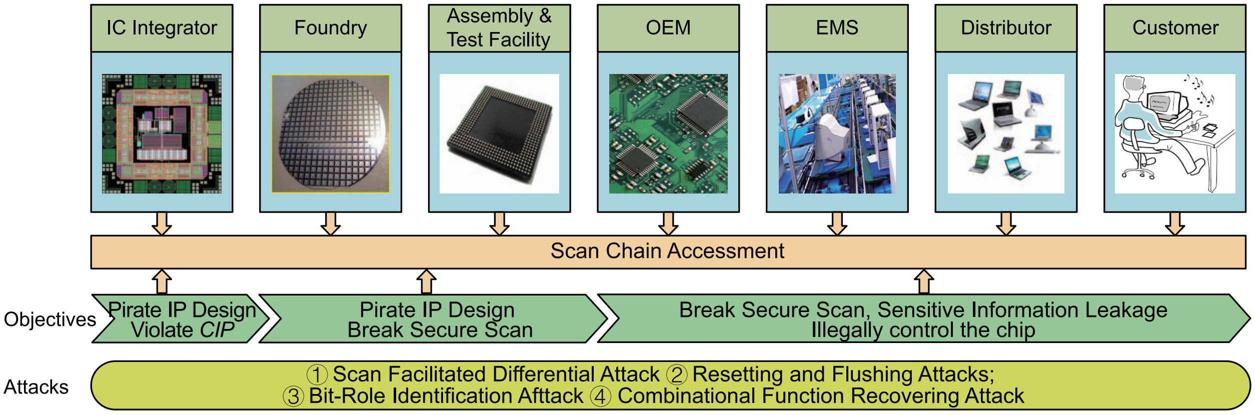
Subobjective 3: The proposed architecture should be able to prevent IP design piracy throughout supply chain.

Subobjective 4: The proposed architecture should be able to prevent the chip from being scanned in illegal values by OEM, EMS, distributor, and end customer.

Furthermore, the proposed architecture should be able to protect full scan, not just critical data registers. As on-chip compression and compaction schemes can usually be bypassed in industrial designs, the proposed architecture should not be bypassable in any test mode. Finally, the proposed architecture should be of low cost, which requires limited impact on the EDA generated DFT scan chain, as well as small area overhead and power consumption.

The assumptions we made when analyzing the proposed architecture as follows.

1) Countermeasures in [44] and [45] which fails reverse engineering can be applied to the proposed architecture in Section III. Thus, only noninvasive scan-based attacks are considered in this paper.


IC Integrator
Foundry
Assembly & Test Facility
OEM
EMS
Distributor
Customer
Scan Chain Accessment
Objectives
Pirate IP Design Violate CIP
Pirate IP Design Break Secure Scan
Break Secure Scan, Sensitive Information Leakage Illegally control the chip
Attacks
① Scan Facilitated Differential Attack ② Resetting and Flushing Attacks;
③ Bit-Role Identification Afftack ④ Combinational Function Recovering Attack

Note: *CIP* = “confidentiality and integrity policies”

Fig. 1. Attacker’s objectives throughout IC supply chain.

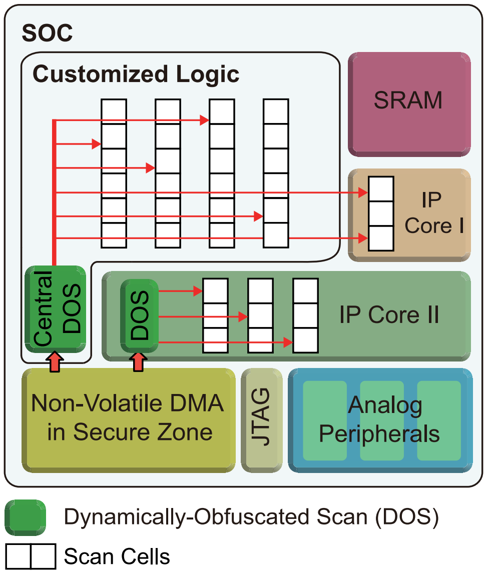


Fig. 2. Overview of an SoC protected by DOS architectures.

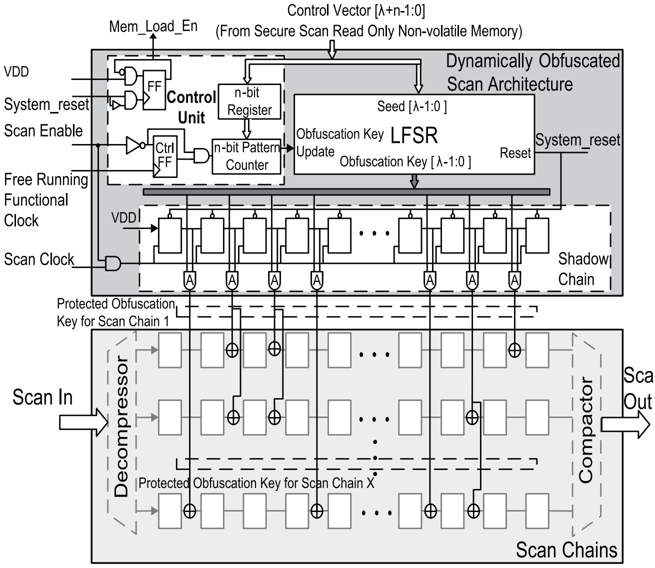


Fig. 3. Detailed architecture of the proposed DOS.

2) The adversary can fully access scan-based test signals, including scan in, scan out, and scan enable.
3) The adversary knows the encryption algorithm used in the circuit by other information sources (i.e., side channel analysis, or other attackers belonging to the IC integrator).
4) The test controller resets all registers when switching from normal to test mode as most industrial chips have adopted it. Thus, attacker has to rely on scan chain for data input and observation.
5) Although on-chip compression and compaction schemes are considered as natural countermeasures, they can usually be bypassed for debugging. Therefore, to verify the security of the proposed architecture, the decompressor and compactor are removed during security analysis.

III. DOS Architecture

The overview of the proposed secure scan in an SoC is shown in Fig. 2. The DOS architecture reads *Control Vector* from nonvolatile directly memory access (DMA) in secure zone, and provides protection to scan chains. The DOS architecture has capacity and flexibility to provide protection for IP owner as well as IC integrator. According to Fig. 2, IP owner can either integrate one DOS into IP, as IP core II, or share the central DOS belonging to the customized logic, as IP core I.

As illustrated in Fig. 3, the proposed DOS architecture is composed of a linear feedback shift register (LFSR), a *Shadow Chain* with XOR gates, and a *Control Unit*.

1) LFSR: The 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 Fig. 3. 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 only when the *Obfuscation Key* update is required. It should be noted that for LFSR, a seed with all zeros is illegal when using an XOR feedback, the LFSR would remain locked-up state, and continues providing all zero *Obfuscation Key*. Therefore, the scan chains cannot be obfuscated. To avoid the above scenario, it is suggested that some of XOR gates in LFSR should be replaced with XNOR gates.

2) Shadow Chain and XOR *Gates:* As shown in Fig. 3, the input of the *Shadow Chain* is the λ-bit *Obfuscation Key* generated by the LFSR, while the outputs are $k\lfloor\lambda \times \alpha\rfloor$-bit *Protected Obfuscation Keys*, where α is the permutation rate (the percentage of bits permuted inside each DFT scan chain), and k is the number of scan chains. The *Shadow Chain* is designed for propagating the *Obfuscation Key* at the ith scan cell along the scan chain when the ith scan clock comes. Therefore, the *Shadow Chain* is able to: 1) protect the *Obfuscation Key* from being leaked through resetting attack; 2) prevent any unscrambled data from being scanned out; and 3) 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 Fig. 3, the data input of its first flip-flop is connected to VDD. The XOR gate inserted after the ith scan cell of Scan Chain X is controlled by the output of the ith flip-flop of the *Shadow Chain* through a *Type A* AND gate. As shown in Fig. 3, the *Type A* AND gates of DOS 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 λth scan clock, the scrambled response starts to show up at the scan output. At the same time, the *Shadow Chain*'s 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 Fig. 3, 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 nonvolatile 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 DOS generates the Mem_Load_En signal. This signal allows the *Control Vector* of DOS 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 DOS at powering up, which can be guaranteed by hard coding the *Control Vector* address in DMA and 2) limited readability: the *Control Vector* can only be read by DOS, which can be satisfied by using the handshaking signal Mem_Load_En (in Fig. 3) generated by DOS, as an input of the DMA address accessing authorization. Additionally, as shown in Fig. 3, during scan, Mem_Load_En also enables the *Control Vector* can only be read once after the reset event. Furthermore, the memory encryption technique such as [46], which allows the *Control Vector* to be stored into the nonvolatile memory in an encrypted manner, is recommended but not required. When the pattern counter value reaches p, the *Obfuscation Key* is updated. Otherwise, the *Obfuscation Key* is locked. As sometimes the set of test patterns cannot be delivered at once, this feature offers the IP owner flexibility to dynamically add new patterns with updated *Obfuscation Key*.

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 *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*.

Fig. 4 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 λth scan clock comes, the scan out starts to output obfuscated responses [waveform (h)].

IV. Test Methodology Based on DOS

This section discusses the secure test methodology based on the proposed DOS within the supply chain.

A. Secure Test Methodology

Scan-based tests are required for wafer, assembly, and sometimes system tests. An overview of the test methodology with DOS is shown in Fig. 5.

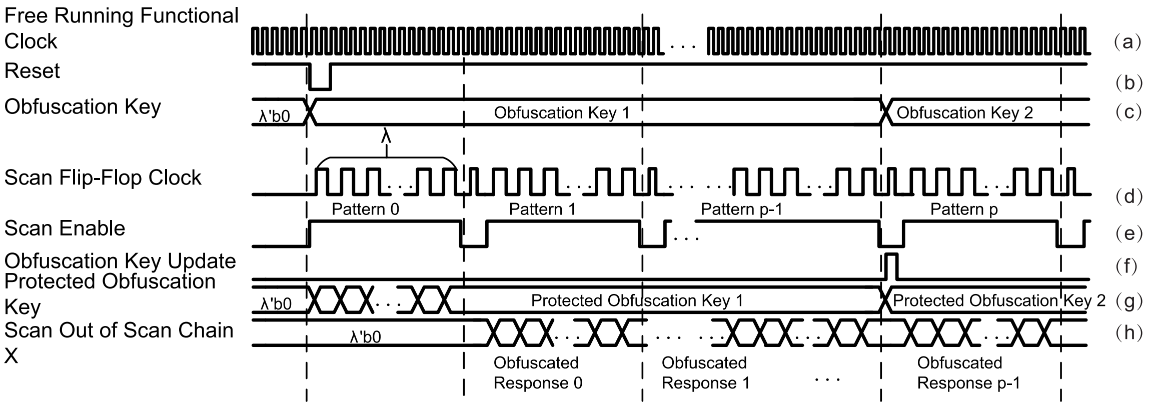


Fig. 4. Timing diagrams for DOS architecture.

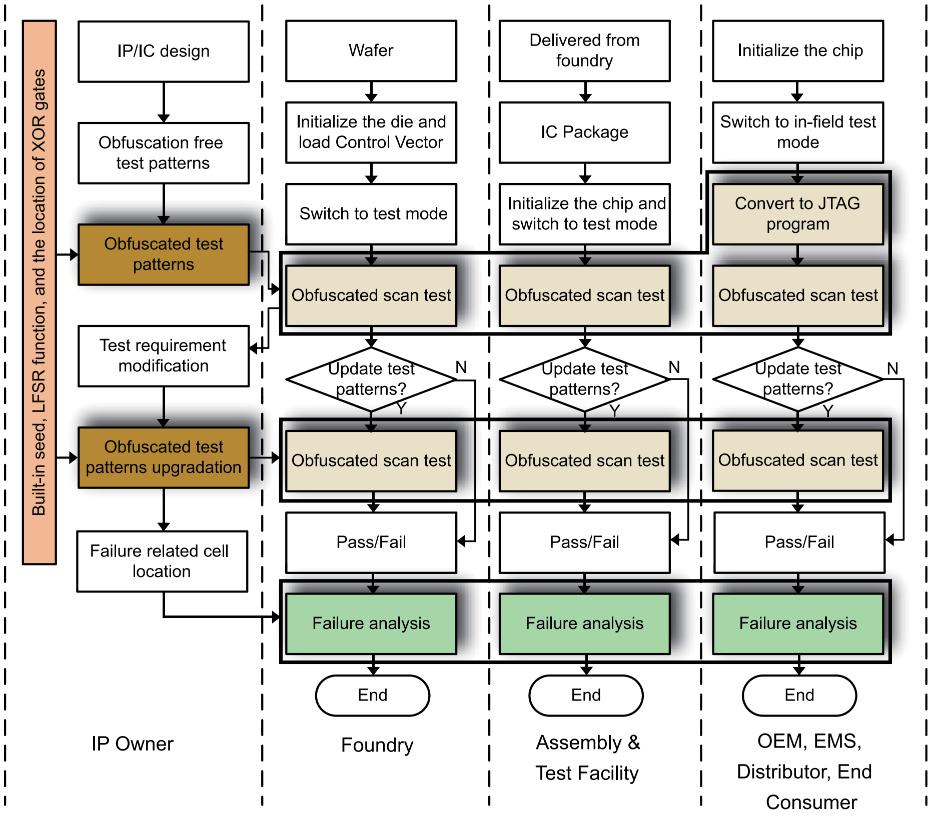


Fig. 5. Secure test methodology based on DOS in the supply chain.

1) At IP Owners: As shown in Fig. 5, 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 DOS inserted netlist, and forcing *Protected Obfuscation Key* as λ'b0. 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 algorithm for obfuscated patterns/responses calculation is shown in Algorithm 1. The obfuscated test patterns and responses will be delivered to testers downstream in supply chain, i.e., IC integrator, foundry, and assembly/test facilities.

2) At Foundry/Assembly: During the first system initialization at foundry, the encrypted *Control Vector* is programmed into the nonvolatile DMA with other 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. As discussed in Section IV-B, 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

Algorithm 1 Obfuscated Patterns/Responses Generation

```
Input: Netlist, built-in seed, LFSR function, and the location of XOR gates
Output: Scrambled pattern ($SP$) & scrambled response ($SR$)
λ is the maximum scan chain length
Original pattern/response = function (Netlist)
    Original pattern $P = \{P_1, P_2, ..., P_\lambda, \}$
    Original response $R = \{R_1, R_2, ..., R_\lambda\}$
$Obfuscation\ Key$ = function (built-in seed, LFSR function)
The scan chain $a = \{a_1, a_2, ..., a_\lambda\}$
if the $i_{th}$ scan cell is obfuscated & $Obfuscation\ Key_i = 1$ then
    $a_i = 1$
else
    $a_i = 0$
end if
Scrambled pattern $SP = \{SP_1, SP_2, ..., SP_\lambda\}$
for $i = 1$; $i < n$; $i++$ do
    $SP_i = P_i \otimes a_1 \otimes a_2 ... \otimes a_{i-1}$
end for
Scrambled response $SR = \{SR_1, SR_2, ..., SR_\lambda\}$
for $i = 1$; $i < n$; $i++$ do
    $SR_i = R_i \otimes a_i \otimes a_{i+1} ... \otimes a_\lambda$
end for
return $SP$, $SR$
```

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.

3) 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 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, as shown in Section IV-B. 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.

B. Flexibly in Updating Pattern/Response

Sometimes test patterns cannot be delivered to foundry, assembly, or in-field debug facilities at once. Or based on the result of the first test, the patterns need to be adjusted, i.e., adding new test patterns, reordering test patterns, or removing some ineffective test patterns. The proposed DOS provides full flexibility for test pattern updating. There are four scenarios as shown in Fig. 6.

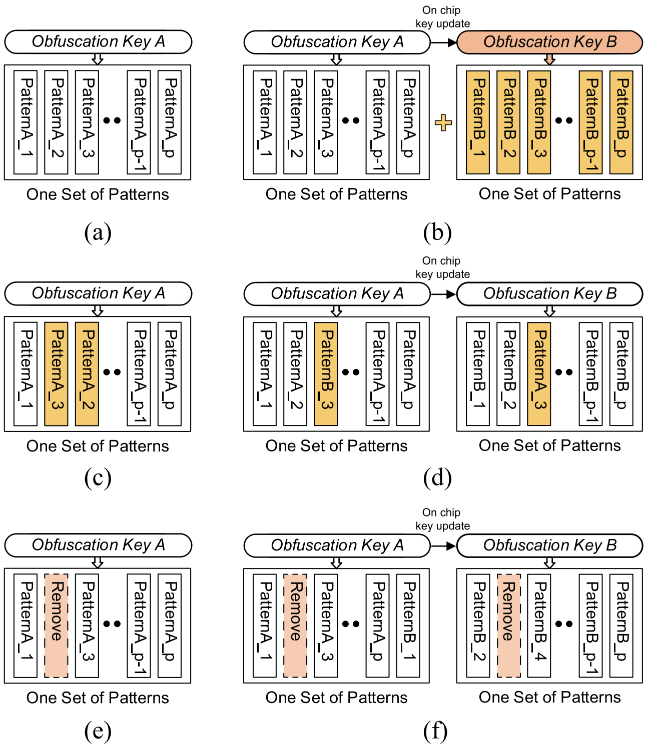


Fig. 6. Schematic for updating test patterns with DOS. (a) Original set of patterns delivered by IP owner with *Obfuscation Key A*. The original set size is *p*. (b) IP owner delivering additional sets of patterns. IP owner delivers the additional set of patterns based on *Obfuscation Key B*. The *Control Unit* automatically updates the *Obfuscation Key A* to *Obfuscation Key B* on-chip. No key delivery is needed. (c) Reordering within one set. The patterns can be reordered freely by test/product engineer. (d) Reordering between two delivered pattern sets. The impacted patterns (PatternA_3 and PatternB_3 in the figure) need to be reobfuscated. IP owner delivers the reobfuscated patterns, and no key update is needed. (e) Deleting any test patterns. Test patterns in a signal set can be deleted by test/product engineer. No input from IP owner needed. (f) Deleting any test patterns within Pattern Set A. Then equaling number of patterns of the Pattern Set B need to be shifted to the Pattern Set A and reobfuscated. No obfuscation needed if any pattern is deleted from Pattern Set B.

1) *Adding Additional Test Patterns:* Sometimes, test engineer or IP owner needs to add additional test patterns for more comprehensive fault detection or failure analysis. As shown in Fig. 6(b), when the original pattern set containing *p* patterns is finished (*p* is determined by IP owner), the on-chip *Control Unit* automatically generates an *Obfuscated Key* updation request. The newly delivered pattern set, which are scrambled with the updated *Obfuscation Key* (*Obfuscation Key B*) by IP owner, can be appended without affecting the original pattern set. Again, there is no key delivery from IP owner required.
2) *Reordering Test Patterns:* The higher power consumption of scan-based testing is a serious concern in the semiconductor industry. Test pattern reordering helps to reduce dynamic power consumption. As shown in [47], test pattern reordering also helps to improve diagnosis

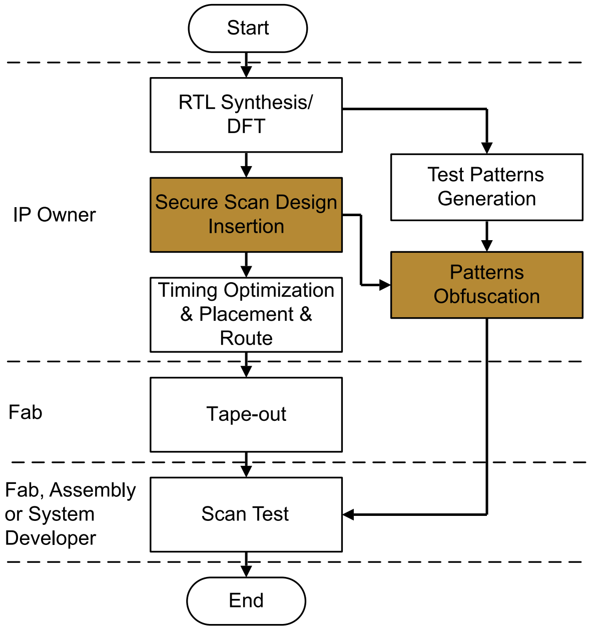

Start
RTL Synthesis/ DFT
IP Owner
Secure Scan Design Insertion
Test Patterns Generation
Patterns Obfuscation
Timing Optimization & Placement & Route
Fab
Tape-out
Fab, Assembly or System Developer
Scan Test
End

Fig. 7. DOS implementation flow. The dark-colored boxes are the impacted steps in the design flow.

resolution. Hence, test pattern reordering frequently happens during production test [48]. Therefore, reordering test pattern with minimum cost is necessary for industrial test. As shown in Fig. 6(a), the DOS updates the *Obfuscation Key* after applying *p* patterns, where *p* is the size of original pattern set determined by IP owner. The order of test patterns within one set can be rearranged freely by test/product engineers at fab or assembly as shown in Fig. 6(c). If the reordering is between different sets, IP owner just needs to reobfuscate the affected patterns [*PatternA*_3 and *PatternB*_3 in Fig. 6(d)] with the corresponding *Obfuscation Keys*.

3) *Deleting Ineffective Test Patterns:* To reduce test cost, low efficiency test patterns of the original pattern set need to be removed. As shown in Fig. 6(e), the pattern removement can be conducted by the test/product engineer. If two pattern sets are delivered at the same time, and any pattern belonging to the previous pattern set (Pattern Set A) is deleted. Then equaling number of patterns belong to the consecutive set (Pattern Set B) need to be shifted to Pattern Set A and reobfuscated which is shown in Fig. 6(f). However, no reobfuscation is needed if any pattern is deleted from the latter pattern set.

In summary, updating the pattern set can be performed with minimum cost, with most of the scenarios being conducted at the test/product engineer side only. It should be noted that the set size *p*, which represents the *Obfuscation Key* updating frequency is determined by the IP owner. A lager *p* reduces the probability of reordering between different pattern sets.

V. DOS Implementation Flow

The DOS implementation flow is shown in Fig. 7 and described below.

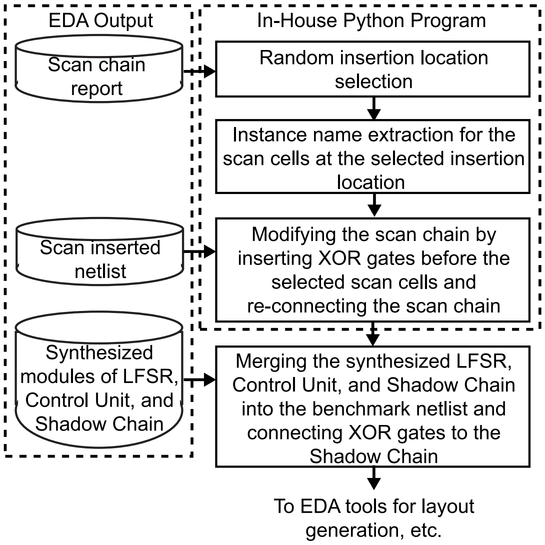

EDA Output
In-House Python Program
Scan chain report
Random insertion location selection
Instance name extraction for the scan cells at the selected insertion location
Scan inserted netlist
Modifying the scan chain by inserting XOR gates before the selected scan cells and re-connecting the scan chain
Synthesized modules of LFSR, Control Unit, and Shadow Chain
Merging the synthesized LFSR, Control Unit, and Shadow Chain into the benchmark netlist and connecting XOR gates to the Shadow Chain
To EDA tools for layout generation, etc.

Fig. 8. Flow for DOS insertion.

Step 1: Design, synthesis, DFT, and test pattern generation. This step is not impacted by the proposed DOS design.

Step 2: DOS insertion and test pattern generation. In this step, the proposed architecture is inserted into the DFT-synthesized design. The flow for XOR gate insertion with the in-house python program and EDA tools are illustrated by Fig. 8. The in-house python program is used to randomly select locations within scan chains for XOR gate insertion, and then generate the TCL scripts for the synthesis tool to insert XOR gates and reconnect scan chains after insertion. Then, the IP owner can generate test patterns (i.e., original and scrambled test patterns) and responses (i.e., original and scrambled test responses) with low computational effort based on the built-in seed and LFSR function. The computation for scrambled patterns and responses is shown in Section IV-A.

Step 3: Placement, routing, and timing closure. In this step, the whole design is placed and routed automatically by EDA tools, and the timing is closed. The LFSR component is distributed in the layout to avoid easy detection.

Step 4: Tape-out and test. In this step, the test patterns/responses are delivered by the IP owner, and the tests are performed by fab, assembly and other testers in supply chain following the test methodology detailed in Section IV. Although the LFSR function is unknown to the test engineer, it is suggested to hide the *Control Vector* into the memory initialization data sequence.

VI. Results and Security Analysis

The proposed technique has been implemented and verified in 32 nm technology node [49] on several benchmark circuits, from Gaisler, OPENCORE, ITC'99, to OpenSPARCT2.

TABLE II
AREA OVERHEAD, SHIFTING POWER OVERHEAD, AND PATTERN PROCESSING TIME OVERHEAD PER PATTERN FOR 10% AND 30% PERMUTATION RATE CASES

Benchmarks		b19	128-bit AES IP	FGU	Leon processor	Leon3s	VGA-LCD
# SFF		6042	2352	27926	1067	17495	17057
# Scan Chains		95	37	437	17	274	268
Area Overhead	10%	0.75%	0.85%	1.05%	1.81%	1.33%	1.41%
	30%	2.00%	2.01%	3.07%	3.57%	3.84%	4.04%
Shifting Power Overhead	10%	0.85%	0.92%	0.90%	0.32%	1.02%	0.83%
	30%	2.50%	1.72%	2.62%	1.72%	2.26%	2.36%
Pattern Proc. Time Per Pattern (μs)	10%	43.7	32.4	86.0	29.6	66.8	62.7
	30%	48.5	37.1	89.3	31.8	76.8	63.0

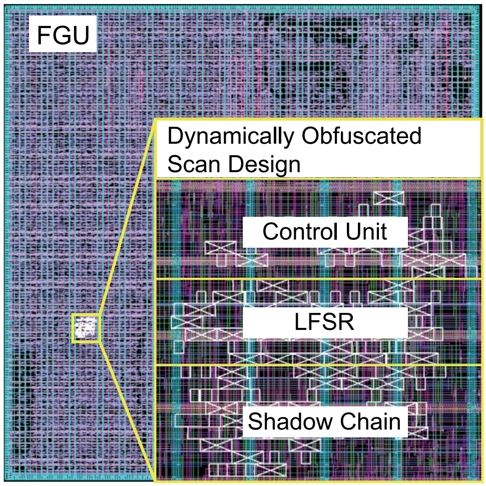


Fig. 9. Layout of DOS protected FGU with 30% permutation rate. The white region includes the cells of a single DOS protecting all scan chains.

The circuits were synthesized with full scan using 100 MHz functional clock and 10 MHz scan clock, and the maximum length of scan chains was 64 ($\lambda = 64$). It should be noted that because the number of scan cells may not be a multiple of 64, EDA tools balance the length of scan chains as closely as possible. Fig. 9 shows the layout of FGU with DOS inserted, and the permutation rate is 30%.

A. Overhead Analysis

As shown in Fig. 3, the overhead mainly comes from the LFSR, the single *Shadow Chain*, and the XOR gates. During implementation, 10% and 30% permutation rates are considered (α = 10%, 30%). From Table II, it can be seen that the area overhead for those two permutation cases are limited to 0.75%–1.81% and 2.00%–4.04%, respectively. The scan shifting power overhead for those two permutation cases are limited to 0.32%–1.02% and 1.72%–2.66%, respectively. As DOS mainly operates and causes power consumption during scan, the shifting power overhead is obtained by comparing the overall shifting power consumption of the original benchmark and DOS inserted circuits. The pattern processing time overhead per pattern due to obfuscation is shown in Table II as well. With a Linux work station of 2.4 GHz, 20-core CPU and a signal thread Python program, the maximum time for patterns obfuscation per pattern is 89.3 μs, which is negligible. The scan chain length's impact on area overhead is shown in Fig. 10. The chain length of the benchmarks is varied from 64 to 96. From Fig. 10, it can be seen that, the area overhead increments, when increasing chain length from 64 to 96, are limited to 0.11%, 0.25%, 0.04%, 0.42%, 0.08%, and 0.07% for b19, 128-bit AES IP, FGU, Leon processor, Leon3s, and VGA-LCD, respectively.

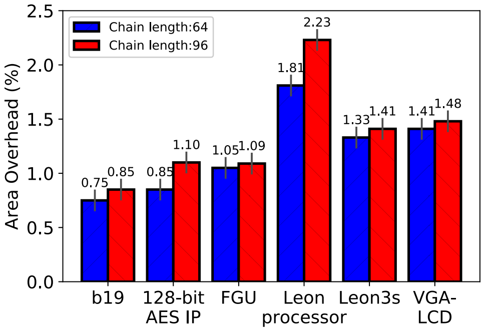


Fig. 10. Chain lengths' impact on area overhead (with 10% permutation rate).

The timings influenced by DOS insertion for different benchmarks are shown in Fig. 11, in which the top 500 critical paths are analyzed. It can be seen that the maximum critical path slack degradation rates caused by DOS insertion are limited to 0.57% (α = 30%) and 0.38% (α = 30%) on average for all benchmarks, which are acceptable. It should be noted that, two separate layouts are generated under 10% or 30% permutation cases. Although the timing constrains are the same for the two cases, the critical paths' layouts are different due to EDA optimization. Thus, the timing of the 30% permutation case are slightly improved for 128-bit AES IP, Leon3s, and VGA-LCD.

B. Security Analysis

The security performance of the proposed architecture will be fully analyzed under existing scan-based noninvasive attacks.

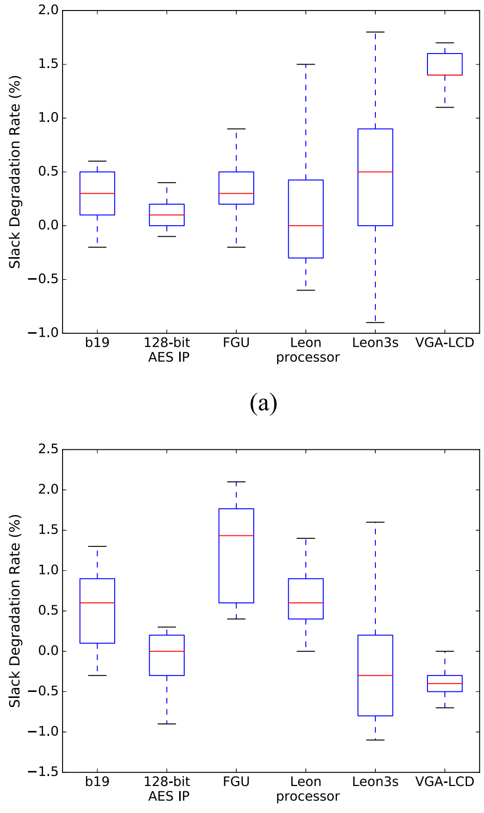
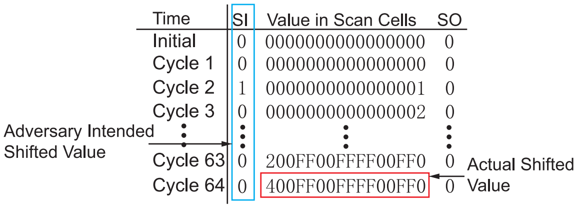


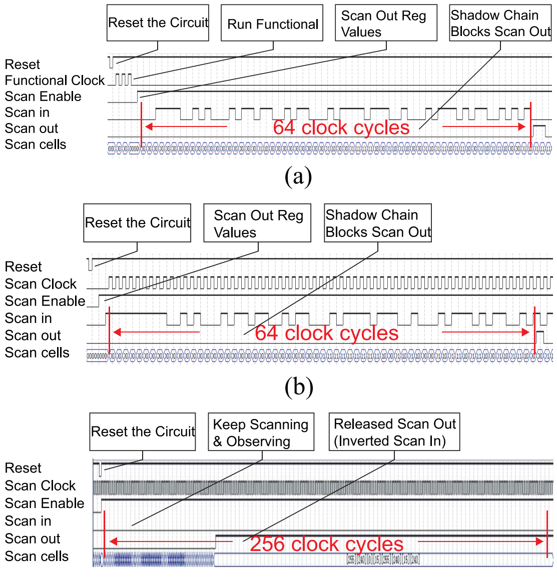
Fig. 11. Timing influenced by DOS insertion for the 500 top paths of different benchmarks. With different permutation rates, the average slack degradation rate are 0.57% ($\alpha = 10\%$) and 0.38% ($\alpha = 30\%$) for all benchmarks. Permutation rate (a) $\alpha = 10\%$ and (b) $\alpha = 30\%$.



Time	SI	Value in Scan Cells	SO
Initial	0	0000000000000000	0
Cycle 1	0	0000000000000000	0
Cycle 2	1	0000000000000001	0
Cycle 3	0	0000000000000002	0
⋮	⋮	⋮	⋮
Cycle 63	0	200FF00FFFF00FF0	0
Cycle 64	0	400FF00FFFF00FF0	0

Adversary Intended Shifted Value
Actual Shifted Value

Fig. 12. Under test mode only differential attack [17], DOS architecture prevents the attacker from shifting in patterns to identify critical bits.



(c)

Fig. 13. Security performance of DOS architecture under attacks. (a) Under normal differential attack, when attacker switches from functional mode to test mode, 64 zeros are scanned out without leaking the intermediate values. (b) Under resetting attack, the first 64 scan out bits are not scrambled, thus the *Obfuscation Key* values cannot be identified. (c) Under flushing attack, original or inverted flushing sequence is shifted out which does not leak the *Obfuscation Key*.

1) Differential Attack: During the differential attack [40], after reset, the attacker first runs the system with several clock cycles in normal mode, and then switches it to test mode to scan out intermediate values and to identify the critical flip-flops. The waveforms of Fig. 13(a) indicates the attackers' inputs to perform a normal differential attack, and the outputs he/she need to observe for this attack, which includes several functional clocks to push functional data into registers (shown in the functional clock waveform), the consecutive scan mode switching (shown in the scan enable waveform), an arbitrary flushing sequence shifted in (shown in the scan in waveform), and the scan out sequence under DOS protection (shown in the scan out waveform). Due to the existence of the *Shadow Chain*, which always blocks the scan out for the first $\lambda = 64$ scan clocks, 64 zeros are scanned out as shown in Fig. 13(a). Hence, no intermediate functional data is leaked under this attack.

2) Test Mode Only Differential Attack: As in most industry designs, scan chains are reset when switching between test and other modes (i.e., resetting countermeasure). Hence, normal differential attack [40] can be defended. However, a new test mode only differential attack has been proposed in [17]. First, this attack needs to shift in all-zero and specific one-hot patterns for secure bit identification. However, with the proposed architecture, due to the fact that the *Obfuscation Key* is protected as analyzed below and the *Shadow Chain* uses the *Obfuscation Key* to scramble all scan in bits, the actual patterns shifted in are not useful for secure bit identification. For example, as shown in Fig. 12, if the adversary intends to shift in 64'h4000_0000_0000_0000, disturbed by the *Obfuscation Key*=64'h0010_1010_0010_1010, the actual value shifted in is 64'h400F_F00F_FFF0_0FF0. Therefore, the proposed design is resilient to the test mode only differential attack.

3) Resetting Attack: After resetting the circuit, the attacker knows the values of all flip-flops (all zeros) before scanning out. Hence, the *Obfuscation Key* risks being leaked through an analysis of the scanned out values. However, as shown in Fig. 3, the *Shadow Chain* of the proposed architecture always blocks the first λ scan out bits after reset. As shown in Fig. 13(b), when the attacker performs resetting attack, $\lambda = 64$ zeros are scanned out, which prevents the attacker from obtaining the *Obfuscation Key*.

4) Flushing Attack: To observe the LFSR sequence, the attacker may select a scan out pin, and keep scanning out the scan chain values. When a stream of bits are flushed in the scan chain without executing any functional clock cycle, the flushing attack discussed in Section II-B is considered to

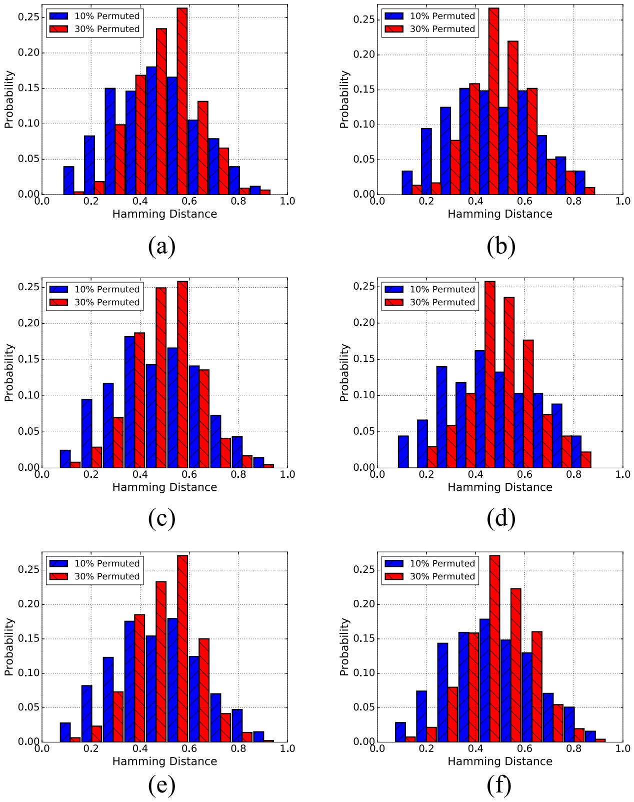
TABLE III
PROBABILITY OF BRUTE FORCE GUESSING THE *Protected Obfuscation Key* UNDER 10% AND 30% PERMUTATION RATES

Chain length (λ)	10% Permuted	30% Permuted
64	$1/2^{31}$	$1/2^{71}$
128	$1/2^{66}$	$1/2^{146}$
256	$1/2^{139}$	$1/2^{296}$

occur. However, as shown in Fig. 3, due to the existence of *Shadow Chain*, for the very first scan in clock cycles, the *Shadow Chain* ensures the bits of the *Obfuscation Key* is synchronously placed along the scan chain with the scan in bits. Therefore, all scan in bits, including the very beginning ones, and all latter ones, experience the same times (equaling to the count of 1 bits in the *Protected Obfuscation Key* generated by LFSR) of inversions. Therefore, depending on whether the total inversion count is odd or even, the original or inverted flushing sequence is shifted out. For example, as shown in Fig. 13(c), if the attacker keeps scanning in 256 bits of 0 into a scan chain, with the *Obfuscation Key* = 64'h0010_0010_0010_1010, which contains odd number of bit 1, the scanned out vector is 256 bits of 1. Therefore, the attacker cannot obtain any LFSR output sequence from this attack. Moreover, the attacker may make the LFSR to keep sending sequence, by flushing the *Obfuscation Key* update signal in Fig. 3. However, the maximum *Obfuscation Key* update count is limited by the *Control Vector*, the LFSR refuses to send sequences after a specified number of pattern sets, unless the whole circuit as well as the *Shadow Chain* are reset, which is the scenario the same as resetting attack.

5) Brute Force Obfuscation Key Attack: If the adversary wants to use brute force to guess the correct *Protected Obfuscation Key*, the overall probability of guessing the *Protected Obfuscation Key* is $1/2^{\lambda+1}$, since there are $\lambda + 1$ potential positions to insert XOR gates for a scan chain with the length of λ bits. The attacker would be able to know whether there are odd or even 1 bits in the *Protected Obfuscation Key* through flushing attack. Hence, the overall probability of successfully guessing the *Protected Obfuscation Key* of one scan chain is $1/(C_{\lambda+1}^{1} + C_{\lambda+1}^{3} + \cdots + C_{\lambda+1}^{\lambda+1}) = 1/(C_{\lambda+1}^{0} + C_{\lambda+1}^{2} + \cdots + C_{\lambda+1}^{\lambda}) = 1/2^{\lambda}$ (assuming λ is even). For $\lambda = 64$ in this application, this probability will be $1/2^{64}$. Furthermore, even if the attacker knows the permutation rate, the probability of guessing one key is $1/(C_{\lambda+1}^{\lfloor \lambda \times \alpha \rfloor} * 2^{\lfloor \lambda \times \alpha \rfloor - 1})$, where $C_{\lambda+1}^{\lfloor \lambda \times \alpha \rfloor}$ is the overall combinations selecting $\lfloor \lambda \times \alpha \rfloor$ positions, and $2^{\lfloor \lambda \times \alpha \rfloor - 1}$ is the possible key for $\lfloor \lambda \times \alpha \rfloor$ bits as calculated by the equation above. From Table III, it can be seen that successfully guessing the *Protected Obfuscation Key* is still impractical. By randomly inserting XOR gates at the selected chain locations and updating the *Obfuscated Key*, the hamming distance between scan out vectors and the captured responses for each scan chain of different benchmarks is shown in Fig. 14. It can be seen that, for most cases, the hamming distances are as high as 0.4–0.6.

6) Combinational Function Recovering Attack: The non-scrambled scan out vectors can be used to reverse engineer the ASIC functional design [10]. Thus, instead of protecting crypto modules only, it is necessary to protect all scan chains. It should be noted that the probability shown in Table III is the probability of guessing the *Protected Obfuscation Key* of a single scan chain. In order to effectively reverse engineer function of the circuit, the attacker needs to correctly guess the *Protected Obfuscation Key* for all of the scan chains. Hence, for a circuit composed of k scan chains, the attacker's probability of success is further reduced to P^k (P is the probability of guessing one scan chain).



(a) (b) (c) (d) (e) (f)

Fig. 14. Hamming distances between scan out and captured responses for different benchmarks at 10% and 30% permutation rates. For most cases, the hamming distance are between 0.4 and 0.6. (a) b19. (b) 128-bit AES IP. (c) FGU. (d) Leon processor. (e) Leon3s. (f) VGA-LCD.

7) Bit-Role Identification Attack: Prior arts, which integrate the *Authentication Key* [28]–[31] into test patterns for test authentication and obfuscation control, are vulnerable to bit-role identification attack as discussed in Section II. The proposed solution removes the *Authentication Key*, and the obfuscation behavior by scan in. Therefore, the bit-role identification attack does not apply to the proposed design.

8) Obfuscation Key Leakage Risk: Built-in seed, LFSR function, and the locations of XOR gates within each scan chain, determine the obfuscation result, therefore, are the major critical information need to be protected. Proved by the above analyses, these critical information cannot be leaked through the existing scan-based attacks. Also as declared in Section II, anti-reverse engineering technologies such as camouflaging [44], [45] should be applied to the DOS related gates, to prevent stealing the LFSR function, and XOR location by reverse engineering. Furthermore, it is suggested to

TABLE IV
COMPARISON OF DOS WITH VIM-SCAN [31], SSTKR [30], AND SCAN INTERFACE ENCRYPTION [23]

Metrics	Vim-Scan [31]	SSTKR [30]	Scan Intf. Encryption [23]	DOS
Security throughout Supply Chain	Unencrypted test responses exposed in supply chain; vulnerable to bit-role identification attack carried by attackers in fab or assembly	Unencrypted test responses exposed in supply chain; vulnerable to bit-role identification attack carried by all attackers	Authenticated parties with encryption key can get the unencrypted responses	Only IP owner knows the actual unencrypted response; robust against bit-role identification attack
System Requirement	NA	NA	At least one crypto core	DMA and test mode protection
In-Field Diagnosis	Scan based diagnosis	Scan based diagnosis	Functional & scan diagnosis	Scan based diagnosis
Brute Force Key Guessing Probability	$1/2^{M*N}$ [(a)]	$1/2^{2q}C^k_{n_{ff}}$ [(b)]	$1/2^k$	$1/2^{\lambda*k}$ [(c)]
Area Overhead	Extra authentication/verification unit	Extra authentication/verification unit, dummy flip-flops, and LFSR	N-R dummy flip-flops for each scan chain [(d)]	LFSR, single short *Shadow Chain*, and XOR gates
Test Time Overhead	Authentication cycles needed	Lengthen scan chain	R clock cycles for the decryption of pattern's initial bits	No extra authentication cycles needed
Pattern Application Flexibility	Test initialization patterns need to be applied in sequence	Patterns need to be applied in fixed sequence	Yse if $k < \lambda$; No if $k > \lambda$ [(e)]	Patterns in the same delivery can be applied in any sequence
Impact on IP Design	Re-synthesis, layout generation, and tapeout needed when there is key leakage or updation; obfuscation control unit insertion; scan chain modification	Re-synthesis, layout generation, and tapeout needed when there is key leakage or updation; obfuscation control unit insertion; scan chain modification	No re-synthesis, layout generation, or tapeout needed when there is key leakage or updation; scan cipher insertion at scan inputs and outputs	No re-synthesis, layout generation, or tapeout needed when there is key leaking or updation; LFSR, *Shadow Chian* insertion; scan chain modification

[(a)] [31] needs 'M' different 'N' bit authentication key vector set;
[(b)] q, k, and n_{ff} represent the number of LFSR bits, key bits, and flip-flops in [30];
[(c)] k and λ represents the number and the length of parallel scan chains for the proposed design [30][23];
[(d)] N is the width of scan cipher, R is the total number of FF in the original circuit modulo N [23];

stored the encrypted seed in nonvolatile DMA inside the secure zone. It should be noted that the major security risk for the nonvolatile memory is tamper attack, in which the attacker decapsulates the chip, and reads stored values by laser scanning or other types of micro probing [50]. As discussed in Section II, this manuscript only focuses on the noninvasive scan-based attacks. However, popular countermeasures including tamper resistance [51] and anti-reverse engineering techniques [52] are recommended to reduce the security risk. However, in some worst conditions, the critical information may be leaked through un-trusted IC integrator from the design house (IP owner). As the seed, LFSR function, and XOR locations of the DOS belonging to DFT, frontend RTL design, and backend engineers separately, the leakage scenarios can be itemized as follows.

1) If the malicious engineer from the design house only knows one of the built-in seed or LFSR function, the probability to guess the *Obfuscation Key* output by LFSR is $1/(2^{\lambda} - 1)$.
2) If both the built-in seed and LFSR function are known by the un-trusted designer, as the XOR gate location is unknown to him/her, the probability to guess the *Protected Obfuscation Key* is $1/C_{\lambda+1}^{\lfloor \lambda\times\alpha \rfloor}$.
3) If the malicious engineer from the design house only knows the XOR locations, the probability to guess the *Obfuscation Key* output by LFSR is $1/(2^{\alpha\lambda} - 1)$.

For the above three cases, with reasonable obfuscation rate α, obtaining the *Obfuscation Key* is beyond the ability of brute force attack.

C. Comparing With Existing Secure Scan Solutions

Three existing techniques, Vim-Scan [31], SSTKR [30], and Scan Interface Encryption [23], which provide full scan protection, have been selected for comparison. The characteristics of the three solutions are shown in Table IV. As demonstrated in the discussions above, the proposed architecture offers the following advantages: security throughout the supply chain, pattern application flexibility, low overhead, and low impact on IP design and test flow. All of this is provided without compromising testability.

VII. CONCLUSION

In this paper, a novel DOS design is proposed to protect IPs against scan-based attacks throughout supply chain. By scrambling the scan in/out vectors, as well as protecting the *Obfuscation Key*, the proposed architecture can defend against existing noninvasive attacks, and prevent critical information from being stolen by attackers throughout the supply chain (foundry, assembly, system developers, etc.). The DOS technique provides a flexible security strategy for modern designs, which has been verified on benchmarks from Gaisler, OPENCORE, ITC'99, and OpenSPARCT2. The proposed technique is design independent while maintaining low area penalty, power consumption, and pattern generation effort.

REFERENCES

[1] S. S. Ali, O. Sinanoglu, and R. Karri, "Test-mode-only scan attack using the boundary scan chain," in *Proc. 19th IEEE Eur. Test Symp. (ETS)*, Paderborn, Germany, 2014, pp. 1–6.

[2] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. New York, NY, USA: Springer, 2011.

[3] J. Dworak and A. Crouch, "A call to action: Securing IEEE 1687 and the need for an IEEE test security standard," in *Proc. IEEE 33rd VLSI Test Symp. (VTS)*, Napa, CA, USA, 2015, pp. 1–4.

[4] D. Zhang, M. He, X. Wang, and M. Tehranipoor, "Dynamically obfuscated scan for protecting IPs against scan-based attacks throughout supply chain," in *Proc. IEEE 35th VLSI Test Symp. (VTS)*, Las Vegas, NV, USA, 2017, pp. 1–6.

[5] D. Mukhopadhyay, S. Banerjee, D. R. Chowdhury, and B. B. Bhattacharya, "CryptoScan: A secured scan chain architecture," in *Proc. 14th Asian Test Symp.*, 2005, pp. 348–353.

[6] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryption standard," in *Proc. Int. Test Conf. (ITC)*, Charlotte, NC, USA, 2004, pp. 339–344.

[7] R. Nara, N. Togawa, M. Yanagisawa, and T. Ohtsuki, "Scan-based attack against elliptic curve cryptosystems," in *Proc. Asia South Pac. Design Autom. Conf.*, 2010, pp. 407–412.

[8] B. Yang, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 2287–2293, Oct. 2006.

[9] N. Ryuta, K. Satoh, M. Yanagisawa, T. Ohtsuki, and N. Togawa, "Scan-based side-channel attack against rsa cryptosystems using scan signatures," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 93, no. 12, pp. 2481–2489, 2010.

[10] L. Azriel, R. Ginosar, and A. Mendelson, "Exploiting the scan side channel for reverse engineering of a VLSI device," Dept. Elect. Eng., Technion Israel Inst. Technol., CCIT Tech. Rep. #897, 2016.

[11] D. Hely *et al.*, "Scan design and secure chip [secure IC testing]," in *Proc. IOLTS*, vol. 4. 2004, pp. 219–224.

[12] O. Kömmerling and M. G. Kuhn, "Design principles for tamper-resistant smartcard processors," in *Proc. USENIX Workshop Smartcard Technol*, vol. 99. 1999, pp. 9–20.

[13] D. Hely, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Secure scan techniques: A comparison," in *Proc. 12th IEEE Int. On-Line Test. Symp. (IOLTS)*, 2006, p. 6.

[14] G.-M. Chiu and J. C.-M. Li, "IEEE 1500 compatible secure test wrapper for embedded IP cores," in *Proc. IEEE Int. Test Conf. (ITC)*, Santa Clara, CA, USA, 2008, p. 1.

[15] L. Pierce and S. Tragoudas, "Enhanced secure architecture for joint action test group systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 7, pp. 1342–1345, Jul. 2013.

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

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

[18] C. Liu and Y. Huang, "Effects of embedded decompression and compaction architectures on side-channel attack resistance," in *Proc. VTS*, Berkeley, CA, USA, 2007, pp. 461–468.

[19] S. M. Saeed, S. S. Ali, O. Sinanoglu, and R. Karri, "Test-mode-only scan attack and countermeasure for contemporary scan architectures," in *Proc. IEEE Int. Test Conf. (ITC)*, Seattle, WA, USA, 2014, pp. 1–8.

[20] J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "Are advanced DfT structures sufficient for preventing scan-attacks?" in *Proc. IEEE 30th VLSI Test Symp. (VTS)*, 2012, pp. 246–251.

[21] J. Da Rolt, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "Scan attacks and countermeasures in presence of scan response compactors," in *Proc. 16th IEEE Eur. Test Symp. (ETS)*, Trondheim, Norway, 2011, pp. 19–24.

[22] A. Das, B. Ege, S. Ghosh, L. Batina, and I. Verbauwhede, "Security analysis of industrial test compression schemes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 12, pp. 1966–1977, Dec. 2013.

[23] M. Da Silva *et al.*, "Scan chain encryption for the test, diagnosis and debug of secure circuits," in *Proc. 22nd IEEE Test Symp. (ETS)*, Limassol, Cyprus, 2017, pp. 1–6.

[24] M. Inoue, T. Yoneda, M. Hasegawa, and H. Fujiwara, "Partial scan approach for secret information protection," in *Proc. 14th IEEE Eur. Test Symp.*, Seville, Spain, 2009, pp. 143–148.

[25] H. Fujiwara, K. Fujiwara, and H. Tamamoto, "Secure scan design using shift register equivalents against differential behavior attack," in *Proc. 16th Asia South Pac. Design Autom. Conf.*, 2011, pp. 818–823.

[26] G. Sengar, D. Mukhopadhyay, and D. R. Chowdhury, "Secured flipped scan-chain model for crypto-architecture," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 11, pp. 2080–2084, Nov. 2007.

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

[28] J. Lee, M. Tebranipoor, and J. Plusquellic, "A low-cost solution for protecting IPs against scan-based side-channel attacks," in *Proc. 24th IEEE VLSI Test Symp.*, Berkeley, CA, USA, 2006, p. 6.

[29] U. Chandran and D. Zhao, "SS-KTC: A high-testability low-overhead scan architecture with multi-level security integration," in *Proc. 27th IEEE VLSI Test Symp. (VTS)*, 2009, pp. 321–326.

[30] M. A. Razzaq, V. Singh, and A. Singh, "SSTKR: Secure and testable scan design through test key randomization," in *Proc. 20th IEEE Asian Test Symp. (ATS)*, New Delhi, India, 2011, pp. 60–65.

[31] S. Paul, R. S. Chakraborty, and S. Bhunia, "VIm-Scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," in *Proc. 25th IEEE VLSI Test Symp.*, 2007, pp. 455–460.

[32] Y. Luo, A. Cui, G. Qu, and H. Li, "A new countermeasure against scan-based side-channel attacks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2016, pp. 1722–1725.

[33] A. Cui, Y. Luo, and C.-H. Chang, "Static and dynamic obfuscations of scan data against scan-based side-channel attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 363–376, Feb. 2017.

[34] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," *IEEE Trans. Depend. Secure Comput.*, vol. 4, no. 4, pp. 325–336, Oct./Dec. 2007.

[35] G. Sengar, D. Mukhopadhyay, and D. R. Chowdhury, "An efficient approach to develop secure scan tree for crypto-hardware," in *Proc. Int. Conf. Adv. Comput. Commun. (ADCOM)*, 2007, pp. 21–26.

[36] K. Rosenfeld and R. Karri, "Attacks and defenses for JTAG," *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 36–47, Jan./Feb. 2010.

[37] G. K. Contreras, A. Nahiyan, S. Bhunia, D. Forte, and M. Tehranipoor, "Security vulnerability analysis of design-for-test exploits for asset protection in SoCs," in *Proc. 22nd Asia South Pac. Design Autom. Conf. (ASP DAC)*, 2017, pp. 617–622.

[38] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Proc. Annu. Int. Cryptol. Conf.*, 1997, pp. 513–525.

[39] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2002, pp. 2–12.

[40] J. D. Rolt, G. D. Natale, M.-L. Flottes, and B. Rouzeyre, "A novel differential scan attack on advanced DFT structures," *ACM Trans. Design Autom. Electron. Syst. (TODAES)*, vol. 18, no. 4, p. 58, 2013.

[41] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing scan design using lock and key technique," in *Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (DFT)*, 2005, pp. 51–62.

[42] J. P. Skudlarek, T. Katsioulas, and M. Chen, "A platform solution for secure supply-chain and chip life-cycle management," *Computer*, vol. 49, no. 8, pp. 28–34, Aug. 2016.

[43] N. Sklavos, R. Chaves, G. Di Natale, and F. Regazzoni, *Hardware Security and Trust*. Cham, Switzerland: Springer, 2017.

[44] M. Shiozaki, R. Hori, and T. Fujino, "Diffusion programmable device: The device to prevent reverse engineering," *IACR Cryptol. ePrint Archive*, vol. 2014, pp. 1–5, Feb. 2014.

[45] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *Proc. Int. Conf. Comput.-Aided Design*, San Jose, CA, USA, 2013, pp. 819–823.

[46] *AMD Memory Encryption*. Accessed: Aug. 5, 2017. [Online]. Available: http://developer.amd.com/wordpress/media/2013/12/AMD_Me mory_Encryption_Whitepaper_v7-Public.pdf

[47] G. Chen, S. M. Reddy, I. Pomeranz, and J. Rajski, "A test pattern ordering algorithm for diagnosis with truncated fail data," in *Proc. 43rd ACM/IEEE Design Autom. Conf.*, San Francisco, CA, USA, 2006, pp. 399–404.

[48] S. Bahukudumbi and K. Chakrabarty, "Test-pattern ordering for wafer-level test-during-burn-in," in *Proc. 26th IEEE VLSI Test Symp. (VTS)*, San Diego, CA, USA, Apr. 2008, pp. 193–198.

[49] (2016). *Synopsys 32/28nm Generic Library*. [Online]. Available: http://www.synopsys.com/Community/UniversityProgram/Pages/generic-libraries.aspx

[50] S. Skorobogatov, "Fault attacks on secure chips: From glitch to flash," in *Proc. Design Security Cryptograph. Algorithms Devices (ECRYPT II)*, 2011, pp. 1–64.

[51] K. Kursawe, A.-R. Sadeghi, D. Schellekens, B. Skoric, and P. Tuyls, "Reconfigurable physical unclonable functions—Enabling technology for tamper-resistant storage," in *Proc. IEEE Int. Workshop Hardw. Orient. Security Trust (HOST)*, 2009, pp. 22–29.

[52] W. M. Clark, J. P. Baukus, and L.-W. Chow, "Memory with a bit line block and/or a word line block for preventing reverse engineering," U.S. Patent 6 459 629, Oct. 1, 2002.

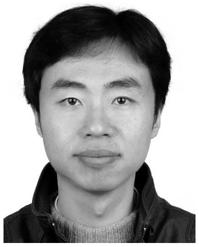
Xiaoxiao Wang (M'08) received the B.S. and M.S. degrees in electrical engineering from Beihang University, Beijing, China, in 2005 and 2007, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA.

She joined the DFT team of Microcontroller Solutions Group, Freescale Semiconductor, Austin, TX, USA, in 2010. She joined the Faculty of Beihang University in 2014. Her current research interests include on-chip measurement architecture design, reliability, and DFT.

Dongrong Zhang received the bachelor's degree from Beihang University, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree.

He is currently with Beihang University, Beijing, researching with Dr. X. Wang. His current research interests include on-chip monitoring, physical design, and on-chip dynamic adaptation methodologies.

Mr. Zhang was a recipient of second and third places of the 25th Feng Ru Cup Competition of Academic and Technological by Inventing "ID Certification System Basing on Personal Movement Identification" and "Color Laser Projection System." He was honorably mentioned by 2015 International Mathematical Contest in Modeling.



Miao (Tony) He (S'16) received the B.S. degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2002 and the M.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006. He is currently pursuing the Ph.D. degree in computer engineering with the University of Florida, Gainesville, FL, USA.

His current research interests include computer-aided design and test for CMOS very large scale integration designs, and hardware security and trust, including test structure, memory built-in self test, test points, secure test, side-channel attacks, and logic encryption.

Donglin Su received the B.S., M.S., and Ph.D. degrees in electrical engineering from Beihang University (BUAA), Beijing, China, in 1983, 1986, and 1999, respectively.

From 1996 to 1998, she finished her Ph.D. researching under a BUAA-UCLA Joint Ph.D. Program in the Electrical Engineering Department, University of California at Los Angeles (UCLA), Los Angeles, CA, USA. She joined the Faculty of the Department of Electrical Engineering, School of Electronics and Information Engineering, BUAA, in 1986, where she is currently a Professor. Her current research interests include numerical methods for microwave and millimeter-wave integrated circuits.

Mark Tehranipoor (S'02–M'04–SM'07) received the Ph.D. degree from the University of Texas at Dallas, Richardson, TX, USA, in 2004.

He is currently the Intel Charles E. Young Professor in cybersecurity with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. He has published over 140 papers, two books, and seven book chapters. His current research interests include computer-aided design and test, reliability analysis, and hardware security and Trust.

Dr. Tehranipoor was a recipient of the Best Paper Award at the 2005 VLSI Test Symposium (VTS), the 2008 North Atlantic Test Workshop (NATW), NATW 2009, and honorable mention for Best Paper Award at NATW 2008, the Best Paper Candidate at the 2006 Design Automation Conference, the Best Panel Award at VTS 2006, the 2008 IEEE Computer Society Meritorious Service Award, and the 2009 NSF CAREER Award. He serves on the program committee of several leading conferences and workshops. He served as the Program Chair for the 2007 IEEE Defect-Based Testing workshop and the 2008 IEEE Defect and Data Driven Testing (D3T), the Co-Program Chair of the 2008 International Defect and Fault Tolerance in VLSI Systems (DFT), and the General Chair for D3T-2009 and DFT-2009. He co-founded IEEE International Workshop on Hardware-Oriented Security and Trust (HOST) and served as HOST-2008 and HOST-2009 General Chair and Chair of Steering Committee. He is currently serving as an Associate Editor for the *Journal of Electronic Testing: Theory and Applications*, the IEEE DESIGN & TEST OF COMPUTERS, and the *Journal of Low Power Electronics*. He is a member of ACM and ACM SIGDA.