ScanSAT: Unlocking Obfuscated Scan Chains

Lilas Alrahis[ϒ], Muhammad Yasin[†], Hani Saleh[ϒ], Baker Mohammad[ϒ],
Mahmoud Al-Qutayri[ϒ] and Ozgur Sinanoglu[‡]
lilas.alrahis@ku.ac.ae, yasin@nyu.edu, hani.saleh@ku.ac.ae, ozgursin@nyu.edu
ϒDepartment of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, U.A.E.
†Tandon School of Engineering, New York University, New York, USA
‡Division of Engineering, New York University Abu Dhabi, Abu Dhabi, U.A.E.

Abstract— While financially advantageous, outsourcing key steps such as testing to potentially untrusted Outsourced Semiconductor Assembly and Test (OSAT) companies may pose a risk of compromising on-chip assets. Obfuscation of scan chains is a technique that hides the actual scan data from the untrusted testers; logic inserted between the scan cells, driven by a secret key, hide the transformation functions between the scan-in stimulus (scan-out response) and the delivered scan pattern (captured response). In this paper, we propose ScanSAT: an attack that transforms a scan obfuscated circuit to its logic-locked version and applies a variant of the Boolean satisfiability (SAT) based attack, thereby extracting the secret key. Our empirical results demonstrate that ScanSAT can easily break naive scan obfuscation techniques using only three or fewer attack iterations even for large key sizes and in the presence of scan compression.

I. Introduction

More and more design houses are going fabless due to the ever increasing cost of Integrated Circuit (IC) manufacturing. Even those who hold onto their fabrication facilities are now outsourcing key steps such as testing [1], [2]. Outsourcing the fabrication and testing processes to potentially untrusted parties raises concerns regarding IC piracy, reverse engineering, overproduction, Intellectual Property (IP) rights violation, and hardware Trojan insertion. Among the design-for-trust (DfTr) solutions developed to prevent such hardware security threats, logic locking is a holistic solution for mitigating IC piracy, Trojan insertion, and overproduction, as it provides protection throughout the IC supply chain.

Logic locking hides the functionality of the design via the insertion of additional logic elements (key gates). The purpose of adding key gates is to lock the circuit during the untrusted phases of the design and manufacturing process. These key gates are driven by key bits (key inputs) that are stored in a tamper-proof memory on the chip. A valid key restores the correct functionality of the design, unlocking it. Logic locking inserts combinational key gates such as XOR/XNORs [3], [4], or multiplexers (MUXes) [5], [6] to lock a design.

In order to provide protection against an untrusted OSAT company during the testing and chip-configuration phases, a special instance of logic locking, namely scan locking, obfuscates the scan chain(s) by inserting key-driven logic in between the Scan Flip-Flops (SFFs). This way, the untrusted tester ends up applying a scan-in stimulus that is different than the pattern delivered into the scan chains; similarly, the tester observes scan-out responses that are different than the captured responses. Scan locking enables a protocol where the designer loads the secret key post-manufacturing on some secure memory. The designer also generates the transformed test data based on the secret key and the original ATPG patterns. The transformed test data is provided to the OSAT company, who performs the testing without knowing the actual ATPG patterns. Similarly, the designer provides transformed configuration vectors that need to be delivered through the scan chains; the OSAT company applies the configuration vectors to customize each part without being able to infer the actual content (security-critical bit streams, chip ID, etc.).

The design flow of a basic scan locking technique is shown in Fig. 1. An example scan locking technique Encrypt Flip-Flop [6], for example, inserts locking MUXes on selected wires (SFFs outputs), producing a locked netlist and Obfuscated Scan Chain (OSC)(s).[1] After fabrication, the chip is activated by inserting the correct key. Even then, the scan chain(s) remain(s) obfuscated from the untrusted tester due to the secret transformations in the scan path.

In this paper, we propose ScanSAT as an attack on OSCs, using Encrypt Flip-Flop as a basic example; we note that ScanSAT can be tweaked and applied to other scan obfuscation techniques as well. The attack flow is presented in Fig. 1. Consistent with almost all attacks on logic locking, the proposed attack requires (i) a working chip (with OSC(s)) and (ii) a locked design netlist. ScanSAT models the OSC as a logic locking problem, and then launches the SAT attack [9][2] on it. It creates the combinational circuit equivalent of the scan-obfuscated circuit; this circuit is a logic-locked circuit with a key corresponding to the secret transformations on the scan chain(s). We then apply SAT attack on this logic circuit

[1]While we use Encrypt Flip-Flop [6] as a representative example, there are also other scan locking variants. While Encrypt Flip-Flop uses statically OSC(s), a Dynamically-Obfuscated Scan (DOS) structure is proposed in [7], where the obfuscation key changes over time. Another similar effort is a Design-for-Security (DFS) architecture proposed in [8] that prevents key information leakage through the scan chain.

[2]SAT attack applies a SAT solver on the CNF representation of the locked netlist to produce a Distinguishing Input Pattern (DIP), which is an input combination for which at least two different key values generate differing outputs. The attack then applies this pattern to the working chip to obtain the correct response, which helps prune all the incorrect keys that fail to produce this output on the locked netlist. This process is repeated iteratively and the generated input-output pairs are added to the gradually growing CNF formulation. The attack succeeds when a DIP can no longer be found by the SAT solver, which is when the correct key is returned.

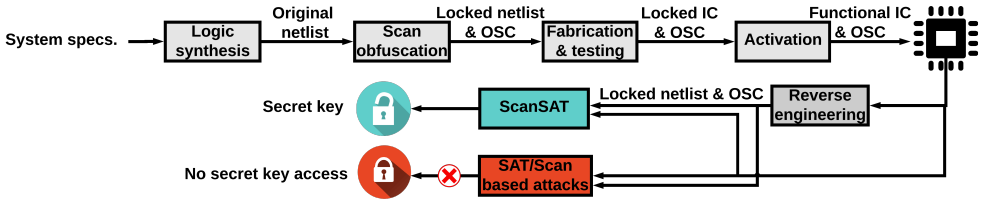

Fig. 1. Deobfuscating basic scan locking using ScanSAT.

model to extract the key; the SAT attack works successfully, simply because basic scan locking techniques do not account for the SAT attack when they embed transformations on the scan path. Our experimental results demonstrate that ScanSAT circumvents basic scan locking (with a single scan chain) within three iterations (i.e., using three or fewer #DIPs).

We also show that ScanSAT easily deobfuscates the scan architectures with multiple OSCs even for large key sizes and in the presence of a scan compression infrastructure.

II. Naive Scan Obfuscation

A. Idea and Implementation

The idea is to obfuscate the scan path via secret inversions; this way, the patterns delivered to the scan chain(s) differ from the scan-in stimulus in a secret manner. In addition, the captured response differs from the scan-out pattern that is observed through the Scan-out pin(s). A secret key dictates these inversion operations on the scan path, and thus, the exact relationship between: (i) the scan-in and delivered patterns and (ii) the captured and scan-out patterns.

Secret inversions can be inserted into the scan chains by inserting XOR gates between the scan cells. An XOR gate driven by a key-bit of 1 implements inversion. A designer can insert k XOR gates, resulting in a k-bit key; some of these k XORs implement inversion on the scan chain. [3]

B. Obfuscated Scan Chain Example

An example OSC is presented in Fig. 2(a) for the $s386$ from the ISCAS-89 benchmark circuits [10]. Three key bits k_0, k_1, and k_2 obfuscate the scan operations. The scan-in pattern applied from the Scan-in pin is denoted by a, where a_i is the bit intended for SFF_i.

In this example, let's assume that the secret key is 111 and thus all three XORs insert inversion. The pattern delivered into the scan cells upon the completion of six shift cycles is denoted by $a^{'}$, where $a_i^{'}$ is the bit delivered into SFF_i. Due to the inversions introduced by the locking XORs, $a \neq a^{'}$. In this example, SFF_2, SFF_3, and SFF_5 receive their stimuli inverted due to an odd number of inversions between the Scan-in pin and these SFFs, while SFF_0, SFF_1 and SFF_4 receive their stimuli as is due to an even number of inversions.

During the capture operation, $a^{'}$ is applied to the combinational circuit rather than a. After the capture cycle, the SFFs will capture their corresponding functional inputs (response of the combinational circuit) denoted by $b_i^{'}$. However, the same locking XORs apply inversions on the response bits as well; the pattern observed through the Scan-out pin is denoted as b, where b_i corresponds to SFF_i. Due to the inversions introduced by the locking XORs, $b \neq b^{'}$. In this example where all XORs insert inversion, the captured response bits in SFF_2, SFF_3, and SFF_5 are observed as is,while response bits in SFF_0, SFF_1, and SFF_4 are inverted prior to being observed through the Scan-out pin.

C. Security Claims

The ever-assumed equivalence of scanned-in to delivered stimuli and captured to observed responses is broken in a secret manner; $a \neq a^{'}$ and $b \neq b^{'}$.

Simple scan-flush attempts by an attacker where special patterns such as all 0's or all 1's are shifted through scan chain(s) with no capture operation reveal very limited information about the secret inversions on the scan path. Such attempts only reveal whether the total number of inversions between the Scan-in pin and the Scan-out pin of the entire chain(s) is even or odd. Information about where on the scan chain(s) these inversions take place remains to be a mystery for the attacker.

III. ScanSAT

In this section, we present the ScanSAT attack on basic scan obfuscation techniques. ScanSAT models the OSC as a logic locking problem and then launches the SAT attack on it. We show that OSCs can be successfully deobfuscated even in the presence of a scan compression infrastructure.

A. Basic Idea

Inspired by modeling the combinational equivalent of a full scan circuit to perform combinational ATPG on it, ScanSAT models the seemingly complex obfuscation of basic scan obfuscation by creating a combinational equivalent of the scan-obfuscated circuit. The obfuscation inversions on the scan path become part of the resultant combinational circuit, which effectively is a logic-locked circuit with key logic inserted at the pseudo-primary inputs/outputs of the circuit; the logic-locked circuit equivalent of a generic scan-obfuscated circuit in Fig. 2(b) is provided in Fig. 2(c), where the obfuscation on the stimulus and the response are modeled separately as combinational blocks driven by the same scan obfuscation key. The resultant circuit can now be attacked via traditional logic-locking attack techniques, such as the SAT attack developed by Subramanyan et al. [9] or the test-data mining attack formulation developed by Yasin et al. [11]. Breaking the logic-locked circuit and extracting its key is equivalent to breaking basic scan obfuscation and deobfuscating/unlocking the scan chain(s).

The proposed ScanSAT modeling is also capable of accounting for any additional logic locking technique applied in conjunction to scan obfuscation; the circuit in Fig. 2(c) allows

[3] Encrypt Flip-Flop adds locking MUXes to achieve the same [6].

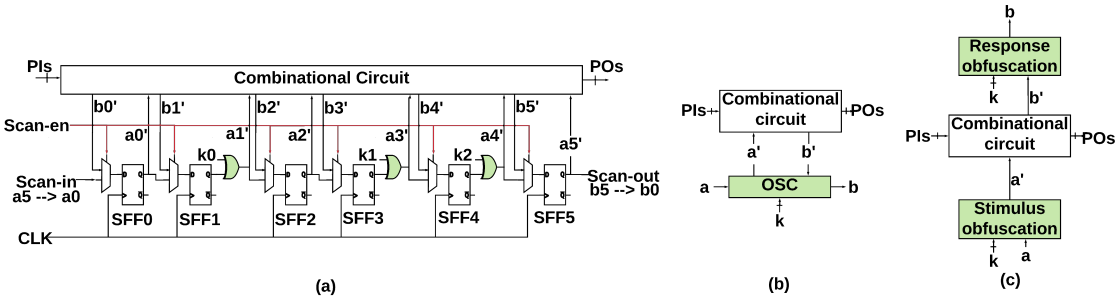

Fig. 2. (a) Scan obfuscation on the $s386$ benchmark; three key bits obfuscate the scan operations. (b) OSC. (c) Modeling the OSC(s) using ScanSAT.

for the incorporation of another logic locking technique (with a separate key) applied on the combinational circuit. In that case, the resultant logic-locked circuit that models two layers of defenses can then be attacked via the SAT attack, extracting both keys simultaneously. We revisit this point in Section D.

Thus, ScanSAT comprises two basic stages: (i) modeling the OSC as a logic locking problem and (ii) breaking the obtained modeled circuit using attacks on logic locking.

The first step is the formulation of the relationship between scan-in pattern a and the pattern delivered into the scan chain(s) $a^{'}$. Some of the bits in $a^{'}$ are identical to the corresponding bits in a while the remaining ones are complementary to the corresponding bits in a. The secret key value k dictates the exact relationship; the bits that pass through an even number of inversions are delivered intact, while those that pass through an odd number of inversions between the Scan-in pin and the target scan cell are inverted. In the example in Fig. 2(a), the following equations capture the relationship between a and $a^{'}$:

$$a_0^{'} = a_0 \tag{1}$$
$$a_1^{'} = a_1 \oplus k_0 \tag{2}$$
$$a_2^{'} = a_2 \oplus k_0 \tag{3}$$
$$a_3^{'} = a_3 \oplus k_0 \oplus k_1 \tag{4}$$
$$a_4^{'} = a_4 \oplus k_0 \oplus k_1 \oplus k_2 \tag{5}$$
$$a_5^{'} = a_5 \oplus k_0 \oplus k_1 \oplus k_2 \tag{6}$$

Without knowing the value of k_i, the attacker cannot tell which stimulus bit is delivered intact and which one is inverted. A capture operation produces the response $b^{'}$ upon applying the scan chain(s) content $a^{'}$ to the combinational circuit. The other obfuscation layer consists of the unknown number of inversions each captured response bit passes through prior to being observed on the Scan-out pin. Again, the values of k_i dictate these secret inversions. The relationship between the captured response pattern $b^{'}$ and the observed scan-out pattern b can be formulated similarly as follows:

$$b_5 = b_5^{'} \tag{7}$$
$$b_4 = b_4^{'} \oplus k_2 \tag{8}$$
$$b_3 = b_3^{'} \oplus k_1 \oplus k_2 \tag{9}$$
$$b_2 = b_2^{'} \oplus k_1 \oplus k_2 \tag{10}$$
$$b_1 = b_1^{'} \oplus k_0 \oplus k_1 \oplus k_2 \tag{11}$$
$$b_0 = b_0^{'} \oplus k_0 \oplus k_1 \oplus k_2 \tag{12}$$

The equations above can be easily modeled as additional XOR logic around the combinational circuit, relating $a^{'}$ to a and b to $b^{'}$ as a function of the secret key k. For the same example, the modeled circuit is shown in Fig. 3. This modeled circuit captures the transformation of inserted scan-in pattern a to the pattern delivered in scan chain(s) $a^{'}$, which is applied through the pseudo primary inputs of the combinational circuit. It also captures the transformation of the captured response pattern $b^{'}$ to the scan-out pattern b. On the inputs side, the same key bit/gate appears multiple times as it affects all the flip-flops to its right on the scan path. Similarly, on the outputs side, the same key bit/gate appears multiple times as it affects all the flip-flops to its left on the scan path. This modeling can also be conceived as unrolling of the scan operations.

The final modeled circuit in Fig. 3 is a logic-locked circuit that has three key bits. An attacker can use this modeled circuit along with the scan-obfuscated oracle to identify the secret key k. For this, the attacker runs the SAT attack [9] on this modeled circuit, generating (obfuscated) input-output patterns[4]. The input patterns a are those the attacker then applies from the Scan-in pin of a working chip. The output patterns b are those the attacker collects from the Scan-out pin of the working chip. By iteratively generating the input-output patterns, the attacker gradually prunes the key search space, and produces the secret key k of the logic-locked circuit, which is also the key used to obfuscate the scan chains.

B. ScanSAT on Scan Compression

The modeling equations developed in the previous section assume that the DfT structure has no scan compression. In this section, we briefly elaborate on how this attack can be extended to scan architectures with scan compression. For simplicity of discussions, and without loss of generality, we utilize fanout decompression and XOR compaction as example stimulus decompression and response compaction, respectively, to explain our attack. We note that the proposed attack can also be applied for any other stimulus decompression and response compaction technique.

With multiple OSCs, the same modeling technique can be applied with no changes. The resulting equations, though, would be simpler than those for the single scan chain case,

[4]Other logic locking attacks, such as sensitization attack, that use a reverse engineered netlist and a working chip can also be used to extract the key from the logic-locked circuit equivalent of OSC(s); in this work, we chose to utilize the SAT attack as it has been the most effective attack on logic locking. It has been shown to break all logic locking techniques that fail to use specific SAT attack resilient structures.

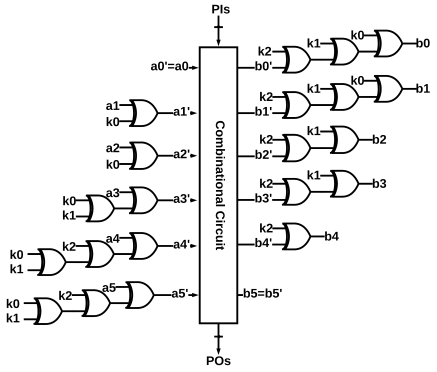

Fig. 3. Modeling the OSC(s) as a logic locking problem.

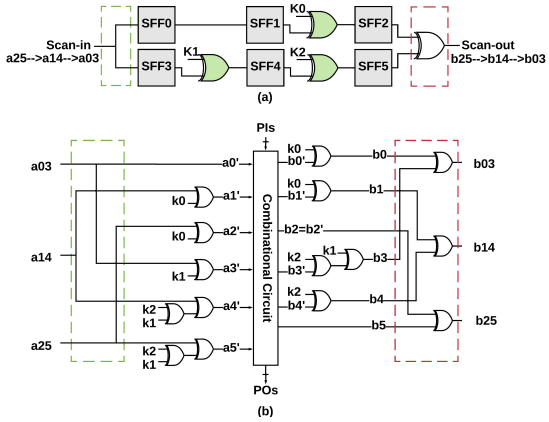

Fig. 4. (a) Applying scan obfuscation on a compression-based architecture; R is 2 and three key bits are used. (b) Modeling as a logic locking problem.

as the cascaded effect of each key bit is limited to the scan cells of the chain where the key bit is inserted.

With stimulus decompressor and response compactor around the OSCs, there is one additional step for ScanSAT modeling; the decompressor and compactor structures need to be instantiated as many times as the number scan slices, capturing the decompression into and compaction of individual slices.[5] The modeled circuit then relates the compressed stimulus to delivered stimulus and captured response to observed (compacted) response, both through key bits. The obfuscated scan architecture with a Compression Ratio (R) of 2 in Fig. 4(a) can be modeled as logic-locked combinational circuit in Fig. 4(b). This modeled circuit shows the fanout decompressor and the XOR compactor each instantiated three times in order to model the stimulus decompression into the three slices and the response compaction of the three slices. The final modeled circuit in Fig. 4(b) is again a logic-locked circuit that has three key bits, which the attacker can break by applying the SAT attack. This time, the input patterns a are compressed scan-in patterns that the attacker applies from the Scan-in pin of a working chip. The output patterns b are the compacted response patterns that the attacker collects from the Scan-out pin of the working chip. The reduced controllability due to stimulus compression and reduced observability due to response compaction may reflect into increased attack difficulty; the computation of the key values is now subject to these controllability and observability challenges. The more aggressive the compression ratio, the more difficult the attack may become.

[5]In scan compression, the group of flip-flops that receive their stimulus in the same shift cycle is referred to as a *scan slice*. The number of scan slices is also referred to as the *scan depth*.

IV. Experimental Results

A. Experimental Setup

In this section, we present the experimental results for ScanSAT attack on several ISCAS-89 benchmark circuits [10] locked using basic scan obfuscation, with Encrypt Flip-Flop [6] used as a representative example. We implemented ScanSAT in a Perl framework on the largest five circuits, i.e., s13207, s15850, s38417, s38584 and s35932. The benchmark circuits equipped with the full scan infrastructure are locked using 32, 64, and 128-bit keys through Encrypt Flip-Flop implementation; specific SFFs are selected for locking as per the selection algorithm discussed in [6].

For launching ScanSAT on the OSCs with compression infrastructure, the largest three benchmarks from the ISCAS-89 (s35932, s38417 and s38584) are locked with a 128-bit key. The SFFs in a design are configured into 16 scan chains and R of 1, 2, 4, 8 and 16 are used.

B. ScanSAT on Naive Scan Locking (Single Scan Chain)

When launched on circuits locked using basic Encrypt Flip-Flop, **ScanSAT is successful in 100% of the cases and retrieves the correct key value for all the circuits across all the key sizes** k. The vulnerability of scan chain obfuscation to the proposed ScanSAT is demonstrated by the fact that **only three or fewer DIPs** are required to unlock the circuits even with k of 128. We attribute this extremely low # of DIPs to the ability of the proposed modeling to efficiently capture the data dependencies in the scan chain. As illustrated earlier in Fig. 3, each key bit affects (i) the stimulus delivered to a scan cell to the right of it and (ii) the response collected from a scan cell to the left of it. Thus, the error introduced by any incorrect key bit is expected to have a unique impact, with the exception of errors being masked during the capture operation. Easy distinguishability of keys results in a very effective key pruning via a SAT attack.

Fig. 5 reports the # of DIPs required to break the locked circuits and the corresponding execution time (on a logarithmic scale) as a function of k. Even with the increasing key size, the attack is able to easily extract the secret key. With increasing key size, the size of the locked circuit modeling the OSC and the associated CNF formula grows linearly, leading to a corresponding increase in the execution time of the attack. For example, the logic locked circuit modeling $s35932$ comprises 65504, 120848, and 231536 gates for k of 32, 64, and 128, respectively; the corresponding execution time is $15.2s$, $25.7s$, and $42.8s$, respectively. The execution time is the highest for $s15850$ circuit for which the # of DIPs is also the highest.

C. ScanSAT on Scan Chain Compression

ScanSAT attack results on scan architectures with scan compression are listed in Table I. Again, the attack is success-

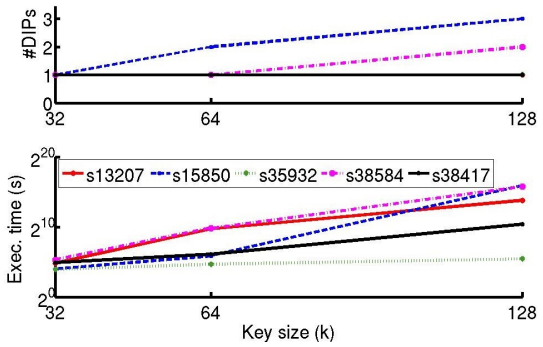

Fig. 5. ScanSAT attack results on a single OSC.

TABLE I
SCANSAT ATTACK RESULTS ON OSCS WITH COMPRESSION. k IS 128 FOR ALL CASES AND 16 SCAN CHAINS ARE CONSTRUCTED IN EACH DESIGN.

Circuit	#SFFs	R	#DIPs	Execution time (s)
s38584	1426	1	2	20.9
		2	1	22.2
		4	2	30.9
		8	1	32.9
		16	1	58.5
s38417	1636	1	1	22.8
		2	1	30.5
		4	3	64.9
		8	2	279.7
		16	1	994.2
s35932	1728	1	1	12.4
		2	1	12.4
		4	1	14.8
		8	1	14.4
		16	1	19.9

ful in 100% of the cases. Comparing the results from Table I with the results presented in Fig. 5, it can be observed that the execution time needed to unlock scan chains is smaller for multiple scan chains; the reason, as mentioned earlier, is the more limited impact of the key bits in case of multiple scan chains. This is further confirmed in Fig. 6, which plots the execution time (on a logarithmic scale) of the ScanSAT attack on the largest three ISCAS-89 circuits as a function of the number of scan chains. Comparing the execution time for a single scan chain vs. two scan chains, for $s35932$, $s38417$ and $s38584$, the attack is $1.8\times$, $1.7\times$ and $638\times$ faster respectively.

The results in Table I also confirm that with scan compression in place, more aggressive compression ratios reflect into increased attack times. The underlying reason, as mentioned earlier, is the reduced controllability and observability.

D. Scan Chain Obfuscation Integrated With RLL

We next investigate the difficulty of breaking scan locking integrated with a combinational logic locking defense, such as Random Logic Locking (RLL), in which XOR/XNOR key gates are inserted at random locations in the combinational circuit [3]. The largest three ISCAS-89 circuits with R of 16 are locked using Encrypt Flip-Flop with k of 128. An additional logic locking layer is integrated with additional 128 key bits, locking the circuits with a total of 256 key

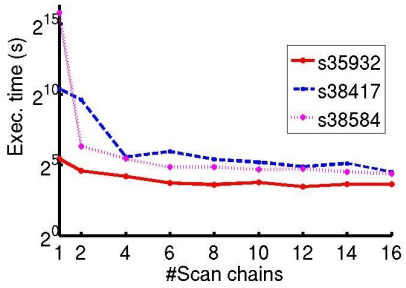

Fig. 6. ScanSAT attack results on multiple scan chains without compression. k is 128 for all cases.

TABLE II
SCANSAT ATTACK RESULTS ON OSCS WITH RLL. R IS 16. TOTAL KEY SIZE IS 256.

Circuit	#DIPs	Execution time (s)
s38584	18	74.0
s38417	41	1216.8
s35932	5	28.1

bits. ScanSAT is then launched on the scan-obfuscated and logic-locked circuits; the results are listed in Table II. The attack is 100% successful on all cases again. Comparing the obtained results with the results presented in Table I, it can be noted that the execution time required to unlock the circuits becomes slightly higher when a second RLL layer is in place. For the $s35932$ circuit, for example, the attack took $1.4\times$ longer time to terminate compared when no RLL was integrated; five DIPs are now utilized by the attack, whereas previously one one DIP was employed. This can be expected as k is now doubled. We conclude that integration of scan obfuscation as another layer of defense over other vulnerable logic locking techniques is still vulnerable to ScanSAT.

E. Comparison Against Attacks With No Scan Access [12]

An orthogonal line of research is an attack that assumes no scan access [12]; herein referred to as NSAA (no scan access attack). Although presented as a decamouflaging attack, the logic locking counterpart of NSAA can be developed. NSAA exercises only the primary inputs (and not the SFFs) and observes only the primary outputs (and not the SFFs) of the working chip. In [12], NSAA is reported to be successful for 80% of the time for a key-size of 32 bits. Often, the attack can correctly retrieve only a subset of key bits. In contrast, ScanSAT works 100% of the time, even on large circuits such as s38584 (even with a key-size of 128). For s38584, the only benchmark common to our and their work, the NSAA tool reportedly crashed [12].

As acknowledged in [12], NSAA is effective only if the ratio of the primary IOs to the SFFs is reasonably large. Unfortunately, this ratio is expected to be small for the realistic circuits as the chip interface (primary IOs) cannot grow at the same rate as the design complexity (SFFs). While NSAA takes on an ambitious goal of attacking with no scan access, such attacks need further development to be successful on realistic designs. In comparison, ScanSAT does not directly access the SFFs either, but rather controls/observes the scan-in/scan-out pin only; the intention of

the scan locking techniques is that obfuscated scan access is the same as no scan access. Nevertheless, we present an attack that is successful consistently, highlighting the inherent vulnerabilities associated with scan locking mechanisms.

V. Discussion

ScanSAT versus scan attacks. ScanSAT is similar to general scan attacks [13]–[16] in the sense that they both utilize the test infrastructure to leak the secret information; however, there are many essential differences between the two types of attacks. The first main difference is regarding the attack objective; while scan attacks aim at extracting the secret key for a publicly known cipher, ScanSAT aims at IP piracy, i.e., retrieving the information hidden in the structure of a netlist. The scan attacks retrieve no structural information since they assume that the knowledge of the cipher is public and that circuit structure does not contain any secret information.

The other difference is in the threat models. ScanSAT and other oracle-guided attacks assume access to a reverse-engineered netlist (IP); such a netlist is not included in the threat model of the scan attacks. As a result, countermeasures for scan attacks are no longer secure in the more generous threat model. Certain scan attack countermeasures rely scan chain authentication [14], [17]–[19]. Access to a reverse-engineered netlist helps circumvent these techniques, as the on-chip logic that implements secure scan can also be reverse-engineered to bypass authentication.

Also, most side-channel attacks rely on operating in normal/user mode for a few cycles and then switching to the test mode; the data loaded into the round registers in the user mode can leak through the scan flip-flops during the test mode. The mode-reset countermeasure (MRC) resets all flip-flops upon transition from one mode to the other and thwarts traditional scan attacks [20]; the countermeasure is deployed in many Intel chips. MRC provides no protection against ScanSAT, as **ScanSAT operates only in the test mode**.

Static vs. dynamic obfuscation. ScanSAT can not only circumvent static scan chain obfuscation for IP piracy, it may also be adapted to leak secret keys for ciphers in the presence of scan chain obfuscation, e.g., in [19], [21]. Since ScanSAT uses only a few DIPs, it may break dynamic scan obfuscation [7], [18] (as utilized in a logic locking threat model) when the required number of DIPs can be applied within one key update cycle.

Limitations of ScanSAT. ScanSAT may fail against Built-in Self Test (BIST) [22] due to the extremely limited observability; this expectation can be verified by extrapolating from the data in Table I for the large compression ratio of BIST.

VI. Conclusion

Obfuscation of scan chains aims to protect against the untrusted testers; naive scan locking techniques obfuscate the scan operations, hiding the relationship between the scan-in and the delivered stimuli and the relationship between the captured and the scan-out responses.

In this paper, we propose the ScanSAT attack on obfuscated scan chains, extracting the secret key and unlocking the circuit/scan chain. The attack is evaluated by analyzing the security of a naive scan obfuscation technique over different scan chain architectures. ScanSAT models the obfuscated scan chains as a logic-locked combinational circuit, paving the way for the application of the powerful SAT attack to reveal the key, unlocking the scan chains, and thus, restoring access to the oracle. We show that ScanSAT can break naive scan locking techniques even for large key sizes and when scan compression is in place.

References

[1] M. Rostami, F. Koushanfar, and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," *IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.

[2] M. Berry and G. John, "Outsourcing Test – What are the most valuable engagement periods?" http://www.amkor.com/go/outsourcing-test, 2014, [May 16, 2016].

[3] J. Roy, F. Koushanfar, and I. L. Markov, "Ending Piracy of Integrated Circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.

[4] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1411–1424, 2016.

[5] J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," *IEEE Transactions on Computer*, vol. 64, no. 2, pp. 410–424, 2015.

[6] R. Karmakar, S. Chatopadhyay, and R. Kapur, "Encrypt Flip-Flop: A Novel Logic Encryption Technique For Sequential Circuits," *arXiv preprint arXiv:1801.04961*, 2018.

[7] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure scan and test using obfuscation throughout supply chain," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[8] U. Guin, Z. Zhou, and A. Singh, "Robust design-for-security architecture for enabling trust in ic manufacturing and test," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2018.

[9] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2015, pp. 137–143.

[10] F. Brglez, D. Bryan, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," in *IEEE International Symposium on Circuits and Systems*, 1989, pp. 1929–1934.

[11] M. Yasin, S. M. Saeed, J. Rajendran, and O. Sinanoglu, "Activation of Logic Encrypted Chips: Pre-test or Post-Test?" in *Design, Automation Test in Europe*, 2016, pp. 139–144.

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

[13] B. Yang, K. Wu, and R. Karri, "Scan based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard," in *IEEE International Test Conference*, 2004, pp. 339–344.

[14] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," *IEEE transactions on dependable and secure computing*, vol. 4, no. 4, pp. 325–336, 2007.

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

[16] S. M. Saeed, S. S. Ali, O. Sinanoglu, and R. Karri, "Test-Mode-Only Scan Attack and Countermeasure for Contemporary Scan Architectures," in *IEEE International Test Conference*, 2014, pp. 1–8.

[17] 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 *IEEE VLSI Test Symposium*, 2007, pp. 455–460.

[18] A. Cui, Y. Luo, and C.-H. Chang, "Static and Dynamic Obfuscations of Scan Data against Scan-based Side-channel Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 363–376, 2017.

[19] Y. Atobe, Y. Shi, M. Yanagisawa, and N. Togawa, "Dynamically Changeable Secure Scan Architecture against Scan-based Side Channel Attack," in *International SoC Design Conference*, 2012, pp. 155–158.

[20] D. Hely, F. Bancel, M.-L. Flottes, and B. Rouzeyre, "Test Control for Secure Scan Designs," in *Test Symposium, 2005. European.* IEEE, 2005, pp. 190–195.

[21] G. Sengar, D. Mukhopadhyay, and D. R. Chowdhury, "Secured Flipped Scan-Chain Model for Crypto-Architecture," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 2080–2084, 2007.

[22] E. J. McCluskey, "Built-in self-test techniques," *IEEE Design & Test of Computers*, vol. 2, no. 2, pp. 21–28, 1985.