SAT-attack Resilience Measure for Access Restricted Circuits

Saran Phatharodom[†], Avesta Sasan[*], Ioannis Savidis[†]

[*]Department of ECE, George Mason University, e-mail: asasan@gmu.edu
[†]Department of ECE, Drexel University, e-mail: {sp694, is338}@drexel.edu

ABSTRACT

With the recent introduction of techniques to restrict scan chain access, a new class of deobfuscation problems emerge, in which the threat model, although similar to deobfuscation of a logic locked circuit, forms a novel class of attack. In this paper, the concept of a logic restricted circuit is generalized and defined. Next, a novel type of SAT-based attack is proposed for the new class of deobfuscation problems, described as a 2-stage SAT-attack. A SAT-attack resilience measure is developed to quantify the security strength of a logic restricted circuit against a SAT-based attack. Finally, the proposed SAT-resilience framework is applied to compare and evaluate effectiveness of example logic restriction schemes.

CCS CONCEPTS

• **Security and privacy** → **Security in hardware**; **Tamper-proof and tamper-resistant designs**; **Hardware security implementation**.

KEYWORDS

hardware obfuscation; logic locking; SAT-attack; security metrics

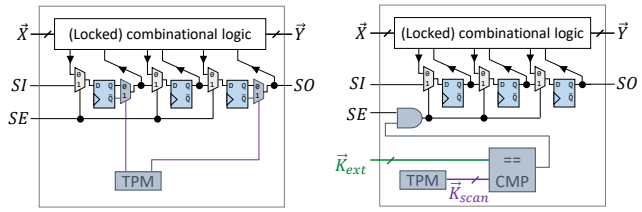
ACM Reference Format:

Saran Phatharodom, Avesta Sasan, and Ioannis Savidis. 2021. SAT-attack Resilience Measure for Access Restricted Circuits. In *Proceedings of the Great Lakes Symposium on VLSI 2021 (GLSVLSI '21), June 22-25, 2021, Virtual Event, USA. ACM, USA, 8 pages.*
https://doi.org/10.1145/3453688.3461759

1 INTRODUCTION

In the past few decades, in order to counter hardware security threats and to operate with untrusted partners along the global supply chain, various types of hardware design-for-trust techniques have been developed including logic locking and IC camouflaging. The two aforementioned families of logic obfuscation techniques were applied first on combinational circuits [1–6], and later, on sequential circuits [7–9]. Among various vulnerabilities including structural/removal attacks [10, 11], bypass attacks [12], and approximate attacks [13, 14], oracle-guided exact-attacks based on Boolean satisfiability formulation (SAT-based attacks) [15, 16] remain as a primary and efficient threat to obfuscated combinational circuits.

For sequential circuits, the SAT-based attacks require different levels of computational resources depending on whether scan chain access is available to the attacker. If scan shift operation is available, the internal states (values of scan flip-flops) are fully controllable and observable, which is as intended for design-for-test purposes. However, given full access to the internal states, the deobfuscation of a locked sequential circuit is reduced to a standard SAT-attack [15, 16] on an obfuscated combinational circuit [17].



(a) (b)

Figure 1: Scan chain access protection using (a) obfuscation (Encrypt Flip-Flop [19]), and (b) access restriction [20, 21].

Without scan chain access, an obfuscated sequential circuit has to be unrolled into multiple time frames as an iterative logic array, after which the regular SAT-attack is applied on the resulting unrolled circuit of combinational logic [18]. SAT-based [18] or model checker-based [17] attacks on sequential circuits require greater computational resources due to the size of the logic array that increases iteratively as the sequential circuit is unrolled into additional time frames.

Consequently, recent work [19–22] has focused on securing access to the scan chain. For example, an encrypted flip-flop [19] is proposed to obfuscate the output of the scan chain by secretly inverting the signal along the scan shift operation as shown in Fig. 1(a). The attacker must know the secret key that sets the location(s) of the signal flip(s) along the chain of flip-flops to correctly interpret the shifted-output sequence. Since the technique does not disable or interfere with the control of the scan enable signal, an attack like ScanSAT [23] is developed that reduces the FF-encrypted sequential circuit into an equivalent key-gate augmented combinational circuit, where the typical SAT-attack [16] is then applied.

As a result, techniques that restrict access of the scan chain have been recently proposed [20, 21]. Instead of obfuscating the output of the scan chain, the techniques aim to disable the scan shift operation. The scan enable signal is stuck at 0 unless the rare correct key is provided as input, where the security mechanism is similar to a password checking for access authorization. The primary components of the technique include the augmentation of an external test-key vector $\vec{K}_{test}$ to the primary inputs and the use of a comparator (CMP) to produce a rare onset signal as a means of authentication. A functionally equivalent circuit of that proposed in [20, 21] is shown in Fig. 1(b).

The development of techniques that restrict access to the scan chain [20, 21] motivates the following questions: (a) Are access restricted circuits the same as logic obfuscated circuits? (b) Are

GLSVLSI '21, June 22–25, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8393-6/21/06…$15.00
https://doi.org/10.1145/3453688.3461759

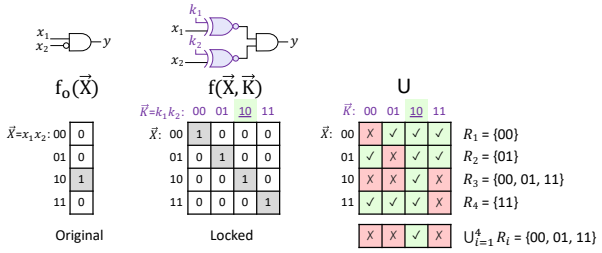


Figure 2: An original circuit f_o and a locked circuit f with the resulting column elimination table U, which demonstrates deobfuscation of f by applying the SAT-attack.

there any other techniques beyond the use of comparator-based topologies that restrict access to the scan chain? (c) What is a consistently accurate way to measure the resilience of an access restricted circuit against a SAT-attack?

The primary goal of this paper is to define and generalize output functionality restriction as a class of hardware security techniques, with application beyond protecting scan chain access, as described in Section 3. A set of attack scenarios and the resulting deobfuscation processes are compared. A novel deobfuscation method is developed that results from analysis of the deobfuscation of access restricted circuits with external key inputs. The new deobfuscation technique requires the execution of the SAT-attack in two stages, which is defined as a "2-stage SAT-attack". The proposed attack and the corresponding framework to measure SAT resilience are described in Section 4.

To further demonstrate methodologies that restrict access to an output, implementations beyond comparator-based locking circuits [20, 21] are investigated in Section 5. Two techniques, one based on out-of-cone SARLock [3] and the other on in-cone XOR/XNOR key-gate insertion [1], are described. The benefits and drawbacks of each restriction methodology are characterized to identify underutilized design considerations that improve the security of the circuit.

In addition, the computational complexity of the 2-stage SAT-attack is proposed as a framework for measuring the resilience of an access restricted circuit against typical SAT-attacks [15, 16]. In this paper, a probabilistic model of the path through the search space taken when executing a SAT-attack [24] is used to analytically evaluate the expected computational complexity, which provides consistent and quantifiable results. Only the basic SAT-attack [16] on combinational circuits is considered. However, the framework to analyze the security of combinational logic is extendable to sequential circuits and to other SAT-based attacks by replacing the theoretical resilience measure used in each step of the 2-stage SAT-attack with the resilience measure of the SAT-based attack of choice.

2 PRELIMINARIES: SAT-ATTACK RESILIENCE OF AN OBFUSCATED CIRCUIT

In this section, basic principles of logic obfuscation, the "SAT-attack" [16], and the measure of SAT-attack deobfuscation effort are described. In this paper, resilience against the SAT-attack is measured analytically as described in [24]. The analytical resilience measure allows for consistent results as compared to the experimentally sampled results that are implementation-specific [24, 25]. In addition, small circuits are chosen as subjects of study rather than larger circuits to allow for the visualization of the entire search space.

2.1 Logic obfuscated circuits

A logic locked circuit $f(\vec{X}, \vec{K})$ hides the correct output functionality $f_o(\vec{X})$ among various other functionalities, where the output functionality is configured by the value of the key signals $\vec{K}$. An example pair of circuits f_o and f are represented as truth-tables in Fig. 2. Only when a correct key pattern $\vec{K}^*$ is applied, does the logic locked circuit f match the output of the original unobfuscated circuit such that $\vec{Y} = f(\vec{X}, \vec{K})|_{\vec{K}=\vec{K}^*} = f_o(\vec{X})$. The correct key vector is secret to the IP owner and is loaded on to an on-chip tamper-proof-memory (TPM) [26]. The uncorrupted output truth-table $U(\vec{X}, \vec{K})$ summarizes the correctness of the output pattern (1:✓or 0:✗) [24].

A second logic obfuscation technique available is IC camouflaging, where camouflaged gates make the functionality of the replaced gate appear ambiguous when image-based reverse engineering of the circuit is performed. When considering the deobfuscation process, an IC camouflaged circuit and a logic locked circuit are equivalent, as any camouflaged gate can be modeled as a key gate [27].

2.2 SAT-attack and SAT-attack search space

Before releasing the logic locked IC to the market, the IP owner loads the correct key to the tamper-proof memory, which activates the locked circuit. The activated live IC now functionally behaves as the intended original circuit and is referred to as the *oracle*. The oracle is also denoted as f_o. For an IC camouflaged circuit, activation is not needed since the fabricated IC already functions correctly. Given a pair of an oracle f_o and the corresponding logic locked version f, a query to the oracle for a specific input pattern yields an input-output pair $(\vec{X}, f_o(\vec{X}))$ observation that eliminates a set of incorrect key candidates. In Fig. 2, a "row set" R_i denotes the set of key patterns or the candidate columns eliminated by the use of an input-output pair corresponding to the input pattern at row i. A collection of sufficient input-output pairs then eliminates all incorrect key candidates and leaves only the correct keys. The deobfuscation process is equivalent to a minimum set cover problem [24, 28]. However, there are many possible sufficient choices of row sets and not all rows are needed.

One efficient process to iteratively select the next input pattern row or distinguishing input pattern (DIP) that is guaranteed to remove at least one new incorrect candidate column not yet removed before, is the SAT-attack [15, 16] due to the use of a Boolean Satisfiability (SAT) solver. The truth-table U captures all possible search paths and is referred to as the U-search space [24]. The search problem of deobfuscating a given pair of f_o and f is the same as solving the resulting U_f search space, where the optional subscript "f" denotes the circuit pair.

2.3 Computational complexity as SAT-resilience measure

From [4, 24], the deobfuscation effort is approximated as a runtime given by $\lambda \times \bar{t}$, where $\bar{t}$ is the average run time per SAT-attack iteration for each DIP used and λ is the number of DIPs used until a correct key is found. As a proxy, the resilience of an obfuscated circuit against a SAT-attack is expressed as λ, the scaling factor [24]. While $\bar{t}$ depends on the netlist topology of the obfuscated circuit f, λ depends on the pattern of (✓/✗) in U, which is independent of the obfuscated circuit topology [24].

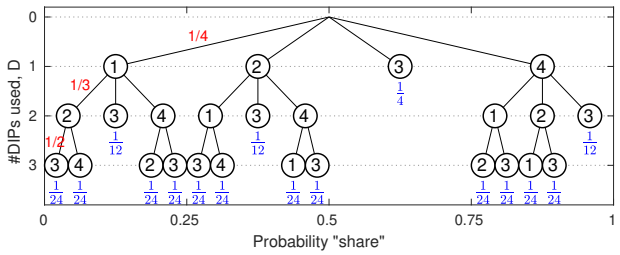


Figure 3: A DIP-sequence tree for a SAT-attack applied to the locked circuit shown in Fig. 2. Assuming an equally likely branching probability, the probability of each branching on the DIP sequence (row 1,2,3) is shown in red, and the probability of the SAT-attack choosing any given DIP sequence is shown in blue directly below the lowest node of each path.

For a given U search space, there are numerous search paths that represent all possible choices of DIP sequences used by the SAT-attack that result in a correct key pattern. The DIP sequence tree for the circuit in Fig. 2 is shown in Fig. 3. Depending on the search space, the search path length, which is the same as the number of DIPs used, possibly varies.

2.4 Probabilistic model for resilience measure

To account for the variation in the search path length, an equally likely branching probability is assumed and a probabilistic model is applied in [24]. Consequently, the number of DIPs used, λ, becomes a random variable denoted as D, which allows for characterization of λ as a probability mass function (PMF). The PMF of D is derived from summing the probabilities of the SAT-attack selecting a search path with the same length. For the DIP-sequence tree shown in Fig. 3, $\mathbb{P}(D = 1) = \frac{1}{4} \times 1 = \frac{1}{4}$, $\mathbb{P}(D = 2) = \frac{1}{12} \times 3 = \frac{1}{4}$, and $\mathbb{P}(D = 3) = \frac{1}{24} \times 12 = \frac{1}{2}$. In this paper, the mean $\mathbb{E}[D]$ (expected resilience), the variance $\mathrm{Var}(D)$, the min D_{min} and the max D_{max} are utilized for comparison of the resilience of different obfuscated circuits.

3 DEFINING LOGIC RESTRICTION

The difference between logic obfuscation and logic restriction is subtle. In this section, logic obfuscation and logic restriction are defined, and the deobfuscation problems that result from an attack on a circuit protected with either of the techniques are described.

3.1 Prior work and proposed classification

To clearly distinguish between the various types of logic-based hardware defense techniques, a classification based on the *output functionality* of the circuit (i.e. the usability of the output by the end user) is proposed. The output function is either correct, undetermined, or incorrect. In the undetermined case, the IC includes some keys that are not yet configured, whereas in the other cases, the IC is fully configured. The three classes are described as (1) logic obfuscation, (2) functionality restriction via logic locking with unconfigured key (logic restriction), and (3) functionality restriction via encryption.

3.1.1 Logic obfuscation encompasses techniques that obfuscate the output functionality of a circuit via logic locking and/or IC camouflaging. By intention, the output of interest behaves correctly for any end user. In other words, the output functionality matches the original circuit output for all input patterns, where $\vec{Y} = f(\vec{X}, \vec{K})|_{\vec{K}=\vec{K}^*} = f_o(\vec{X})$.

3.1.2 Functionality restriction via logic locking with unconfigured keys (logic restriction) are techniques that apply logic locking but include a subset of key bits left unassigned that are to be loaded/inputted by the authorized end user. The output of the subcircuit of interest is not activated. Like logic locking, the user must apply a correct key in order for the circuit to function correctly, where $\vec{Y} = f(\vec{X}, \vec{K}_{ext}) = f_o(\vec{X})$ only if $\vec{K}_{ext} = \vec{K}^*_{ext}$. Therefore, the use of the circuit is restricted as the outputs are corrupted for a subset of input patterns when an incorrect key is applied. Unlike logic obfuscation techniques, by intention, the output behaves correctly only for selected (authorized) users and incorrectly otherwise. In this paper, the class of techniques based on functionality restriction through logic locking with unconfigured keys are simply referred to as *"logic restriction"*. The correct configuring values of the unassigned external key $\vec{K}_{ext}$ are stored on a one-time-programmable tamper-proof memory or a more typical rewrite-able memory such as a shift register. The comparator-based scan chain restriction techniques described in [20, 21] are considered as logic restriction techniques. In Fig. 1(b), the output from the scan chain, *SO*, is considered *logic restricted*.

3.1.3 Functionality restriction via encryption are techniques that produces output signals that are not readily usable, where $\vec{Y} = f(\vec{X}) \neq f_o(\vec{X})$. The output signals must be further decrypted, transformed, or corrected before the expected output results. If logic locking is implemented, the output is already configured. The loaded key value and the camouflaged gate configuration are considered the secret key of the encryption scheme. For example, techniques that obfuscate the scan chain, such as encrypyt FF [19] shown in Fig. 1(a), are considered as functionality restriction via encryption methods. The output from the scan chain, *SO*, is shifted-out through the encrypted FFs, with some values inverted along the chain, while the expected *SO* is the uninverted scan FF values. The authorized user is aware of where the signals are flipped along the chain (the secret) and restores the scan FF values from the "encrypted" *SO* sequence.

3.2 SAT-attack scenarios and resulting deobfuscation processes

In this subsection, scenarios that consider the attacker and the corresponding defense technique are analyzed to demonstrate the subtle differences in the resulting deobfuscation processes. For SAT-attacks on logic obfuscated combinational circuits [15, 16], the threat model consists of two required components: (1) the extracted obfuscated netlist $f(\vec{X}, \vec{K})$, and (2) the active oracle IC $f_o(\vec{X})$. The extracted netlist is acquired through either reverse engineering using imaging based techniques or is leaked from an untrusted partner in the manufacturing and supply chain. For camouflaged circuits, the oracle is simply the fabricated IC. For logic locked circuits, however, the oracle is the activated IC with one of the possibly multiple correct key patterns loaded. In both cases, the oracle is bought from the commercial market.

Consider an attacker A_{maf} (modify-at-fab), an untrusted partner fabricating the IC that wants to steal the IP and modify the obfuscated fabrication mask or the gate level netlist by removing the tamper proof memory and the key gates (and/or camouflaged gates)

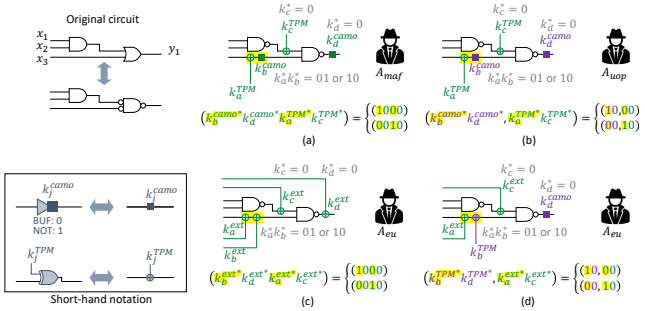


Figure 4: SAT-attack scenarios for attackers deobfuscating logic obfuscated circuits (a)-(b) and logic restricted circuits (c)-(d).

with typical standard cells. For example, for the scenario shown in Fig. 4(a), the objective of attacker A_{maf} is to solve for a correct key pattern and replace the key gates or camouflaged gates with corresponding inverters (INV) or buffers (BUF). For the example, there are two correct key sequences, with the attacker having the option to choose either key.

Next consider another attacker, A_{eu}, an unauthorized end user that wants access to restricted outputs that require the correct external key, as shown in Figs. 4(c) and 4(d). In the scenario shown in Fig. 4(c), the deobfuscation process follows the standard SAT-attack. The threat model requires the activated (de-restricted) live IC acting as the oracle and an extracted restricted netlist. Again, any of the two correct key patterns are valid inputs. However, for the scenario shown in Fig. 4(d), k_b^{TPM} and k_d^{camo} are fixed values that are picked by the defender (IC designer). The TPM has a correct value $k_b^{TPM^*}$ pre-loaded before release to the market, whereas the configuration of the secret, $k_d^{camo^*}$, for a camouflaged IC is fixed upon fabrication. Following a regular SAT-attack, the attacker deduces that there are again two correct key patterns. However, unlike scenario 4(c), in 4(d) there is only one eventual correct pattern that depends on the value of k_b^{TPM} chosen by the defender. The regular SAT-attack is not enough to conclude a correct external key as the pre-defined key value that the attacker cannot change and choose is not yet resolved.

Similarly, in scenario 4(b), another attacker A_{uop} (unlock-over produced-IC) who wants to unlock a not-yet-activated overproduced logic locked IC bought from the black market, faces the same challenge due to the camouflaged gates. Despite the circuits in scenario 4(a) and 4(b) being the same, the resulting deobfuscation processes are different as the controllability of key signals is dissimilar depending on the objectives of each attacker.

From the described attack scenarios, the following are observed: (i) The deobfuscation process of a *logic restricted* circuit is not necessarily different from the deobfuscation process of a *logic obfuscated* circuit, where regular SAT-based attacks are applicable. (ii) Implementing both IC camouflaged gates and key-based logic locking increases the complexity of the deobfuscation process when unlocking an overproduced IC. Similarly, the use of external keys along with camouflaged gates and/or internal preset key bits (loaded in TPM) increase the complexity of the deobfuscation process. In both cases, the presence of unchangeable pre-loaded keys or the use of camouflaged gates increases complexity as the attacker cannot control or modify the key bits or gate configurations.

3.3 Two types of deobfuscation scenarios

Given the same obfuscated and/or restricted circuit, the attack scenario depends on who the attacker is and the purpose of the attack. Ultimately, the important factor is whether or not the attacker is able to control all of the key bits. Let $\vec{K}_c$ denote *controllable* key bits and $\vec{K}_u$ denote *uncontrollable* key bits. For an end user attacker, A_{eu}, that is deobfuscating a logic restricted circuit, the pre-loaded key bits stored in TPM and the configuration of the camouflaged gates are both considered part of the uncontrollable key vector $\vec{K}_u$, whereas the external key bits are considered part of the controllable key vector $\vec{K}_c$. Consequently, two possible deobfuscation processes arise that depend on whether or not $\vec{K}_u$ is present in the extracted netlist. The two cases are described in Sections 3.3.1 and 3.3.2.

3.3.1 Deobfuscate $f(\vec{X}, \vec{K}_c)$: Given $(f_o(\vec{X})f(\vec{X}, \vec{K}_c))$, find $\vec{K}_c$ such that $\forall \vec{X}$, $f(\vec{X}, \vec{K}_c) = f_o(\vec{X})$. All keys are controllable, which allows for the execution of standard SAT-based attacks. Two assumptions are made for the considered threat model: (1) the activated live IC (oracle) is available, where $f(\vec{X}, \vec{K}_c)|_{\vec{K}_c=\vec{K}_c^*} = f_o(\vec{X})$, and (2) the obfuscated/restricted netlist $f(\vec{X}, \vec{K}_c)$ is extracted through reverse engineering or theft.

3.3.2 Deobfuscate $f(\vec{X}, \vec{K}_c, \vec{K}_u)$: Given ($f_o(\vec{X})$, $f(\vec{X}, \vec{K}_c, \vec{K}_u)$, $f(\vec{X}, \vec{K}_c, \vec{K}_u)|_{\vec{K}_u=\vec{K}_u^*}$), find $\vec{K}_c$ such that $\forall \vec{X}$, $f(\vec{X}, \vec{K}_c, \vec{K}_u)|_{\vec{K}_u=\vec{K}_u^*} = f_o(\vec{X})$. A mix of controllable and uncontrollable key bits requires a novel deobfuscation procedure of the circuit, which is more computationally costly to solve. Three assumptions are considered for the given threat model: (1) the activated live IC (oracle) is available, where $f(\vec{X}, \vec{K}_c, \vec{K}_u)|_{\vec{K}_c=\vec{K}_c^*, \vec{K}_u=\vec{K}_u^*} = f_o(\vec{X})$, (2) the obfuscated/restricted netlist $f(\vec{X}, \vec{K}_c, \vec{K}_u)$ is extracted through reverse engineering or theft, and (3) the unactivated live IC given by $f(\vec{X}, \vec{K}_c, \vec{K}_u)|_{\vec{K}_u=\vec{K}_u^*}$ is available.

4 SAT-ATTACK RESILIENCE MEASURE OF A LOGIC RESTRICTED CIRCUIT

The deobfuscation problem described in Section 3.3.2 requires a novel attack methodology, which is described as a 2-stage SAT-attack. The computational complexity of the 2-stage SAT-attack is proposed as a resilience measure of a logic restricted circuit.

4.1 2-stage SAT-attack algorithm

After applying a standard SAT-attack, an additional step is needed to solve for the uncontrollable key vector $\vec{K}_u^*$ as a selection between multiple correct keys must be made, which is shown by the scenarios represented in Fig. 4(b) and 4(d). To resolve $\vec{K}_u$, a second SAT-attack is launched, but with the oracle being the unactivated live IC. Let $f(\vec{X}, \vec{K}_u\vec{K}_c)$ denote the same extracted obfuscated/restricted netlist but with the two key vectors concatenated together as a single vector. To conform to the standard SAT-attack structure, the extracted netlist "$f(\vec{X}, \vec{K})$" is provided as input with two sets of arguments that include the primary inputs (PI) and the key vectors. Let $h(\vec{K}_c\vec{X}, \vec{K}_u)$ be the same extracted netlist but with the grouping of $\vec{K}_c$ as part of the argument of PIs, which leaves only $\vec{K}_u$ as an argument of keys. The second stage attack is then the SAT-attack deobfuscation of $h(\vec{K}_c\vec{X}, \vec{K}_u)$ with the oracle being the

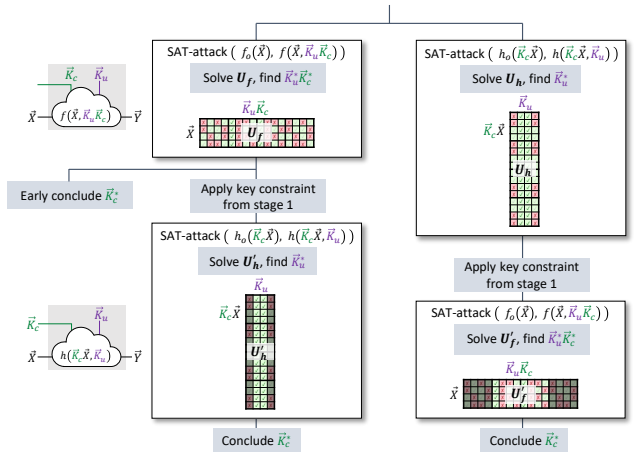


Figure 5: Overview of 2-stage SAT-attack.

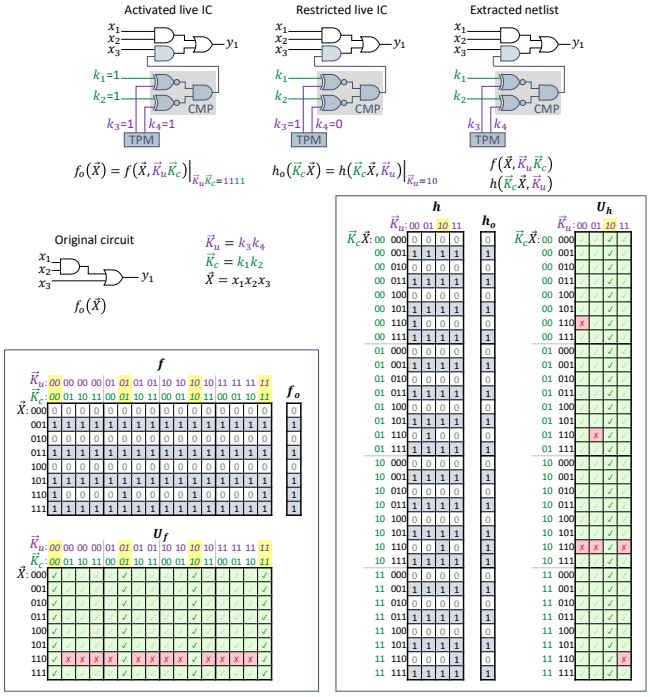


Figure 6: Two deobfuscation subproblems of 2-stage SAT-attack: solve U_f and solve U_h.

unactivated live IC itself, $h(\vec{K}_c\vec{X},\vec{K}_u)|_{\vec{K}_u=\vec{K}_u^*} \equiv h_o(\vec{K}_c\vec{X})$. Therefore, the execution of the SAT-attack in two stages is identified as two deobfuscation subproblems that consist of (1) solving the U_f search space for a given (f_o, f) and (2) solving the U_h for a given (h_o, h). The application of the SAT-attack in two stages is shown on the left path of the overview diagram depicted in Fig. 5. To demonstrate the derivation of the associated search spaces U_f and U_h, an example of a simple secured digital circuit is shown in Fig. 6. Since $h_o(\vec{K}_c\vec{X}) = h(\vec{K}_c\vec{X},\vec{K}_u)|_{\vec{K}_u=\vec{K}_u^*}$, the truth-table of the unactivated live IC (the oracle) h_o varies depending on the value of $\vec{K}_u^*$ chosen by the defender. Consequently, the (✓/✗) pattern of U_h depends on the chosen secret $\vec{K}_u^*$. Note that although there are possibly multiple correct $\vec{K}_u^*$, exactly one pattern is realized for a given live IC under the given attack model.

Alternatively, another approach is to solve U_h for $\vec{K}_u^*$ first, then solve U_f for $\vec{K}_u^*\vec{K}_c^*$ before resolving $\vec{K}_c^*$, as shown on the right path of Fig. 5. The order of execution between U_f-first and U_h-first potentially affects the total number of DIPs needed to resolve $\vec{K}_c^*$ once key constraints determined from the first SAT-attack are applied to the second stage, as described in Section 4.3. Note that the final goal is to find a correct controllable key $\vec{K}_c^*$, and that resolving the correct uncontrollable key $\vec{K}_u^*$ is not needed.

4.2 Possible early conclusion with unique or weak multiple correct key patterns

After solving for U_f by executing the first SAT-attack, in some cases, $\vec{K}_c^*$ is already resolved without the need for the execution of the second SAT-attack on U_h. Consider, for example, circuits 2 through 4 shown in Fig. 7, where there are two correct key patterns. The three circuits are identical except for the selection of which key bits are stored in the TPM and which key bits are externally applied. For circuit 2 shown in Fig. 7, the two correct key patterns $(\vec{K}_u^*, \vec{K}_c^*)$ are $(00, 01)$ and $(00, 10)$. Although, there are multiple correct key patterns, the attack is conclusive as $\vec{K}_u^*$ is unique (00). Therefore, the attacker simply chooses either $\vec{K}_c$ = 01 or 10 to activate/de-restrict the output. Similarly, for circuit 4, although there are two different $\vec{K}_u^*$, the only unique $\vec{K}_c^*$ is 00. The attack again concludes with the execution of a single stage of the standard SAT-attack. For circuit 1, there is only one correct key, $\vec{K}_c^*$ is determined and the attack concludes by just solving U_f.

However, for circuit 3, both vectors $\vec{K}_u^*$ and $\vec{K}_c^*$ include multiple correct key sequences, which forces the attacker to choose $\vec{K}_c^*$ depending on the secret value of the uncontrollable key $\vec{K}_u^*$. A single execution of the SAT-attack to solve for the U_f search space alone is not sufficient and requires an additional execution of the SAT-attack to determine the U_h search space as well.

As a quick example, suppose a circuit has a set of correct keys of $S_{(\vec{K}_u^*,\vec{K}_c^*)} = \{(100, 00), (100, 01), (100, 10), (111, 10)\}$. By strategically choosing $\vec{K}_c$ = 10, the end user attacker is guaranteed to activate the restricted output, regardless of whether the defender sets $\vec{K}_u$ to 100 or 111. Therefore, after executing the first stage of the SAT-attack to solve for U_f, a check to confirm rather the termination condition is met provides benefit as execution of the second stage of the SAT-attack to solve for U_h may no longer be needed.

4.3 Transferring constraint on key-space from first stage of SAT-attack

After solving for U_f, the constraint on the joint key $\vec{K}_u^*\vec{K}_c^*$ is determined. Constraints on the correct key must be utilized to reduce the search effort when executing the second stage of the SAT-attack when solving for U_h. One approach is to remove the other columns from the U_f space and keep only the columns with $\vec{K}_u$ pattern that are in the set of correct uncontrollable key patterns $S_{\vec{K}_u^*}$ found from execution of the first round of SAT-attack on U_f. With respect to the U_h search space, graphical representation of the eliminated columns are shown in black shade in Fig. 8. To distinguish from the original U_h search space, let U'_h denote the search space with

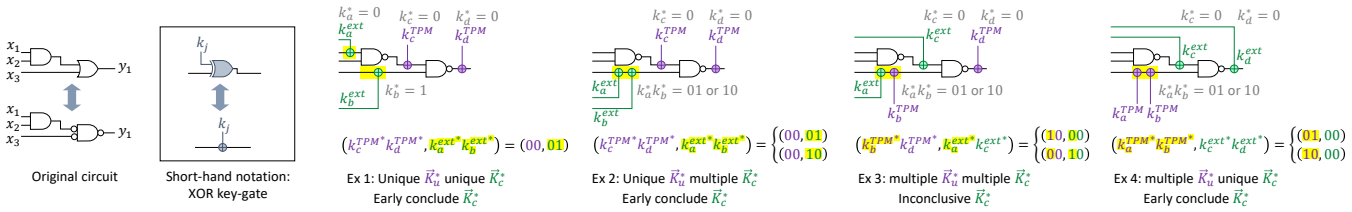
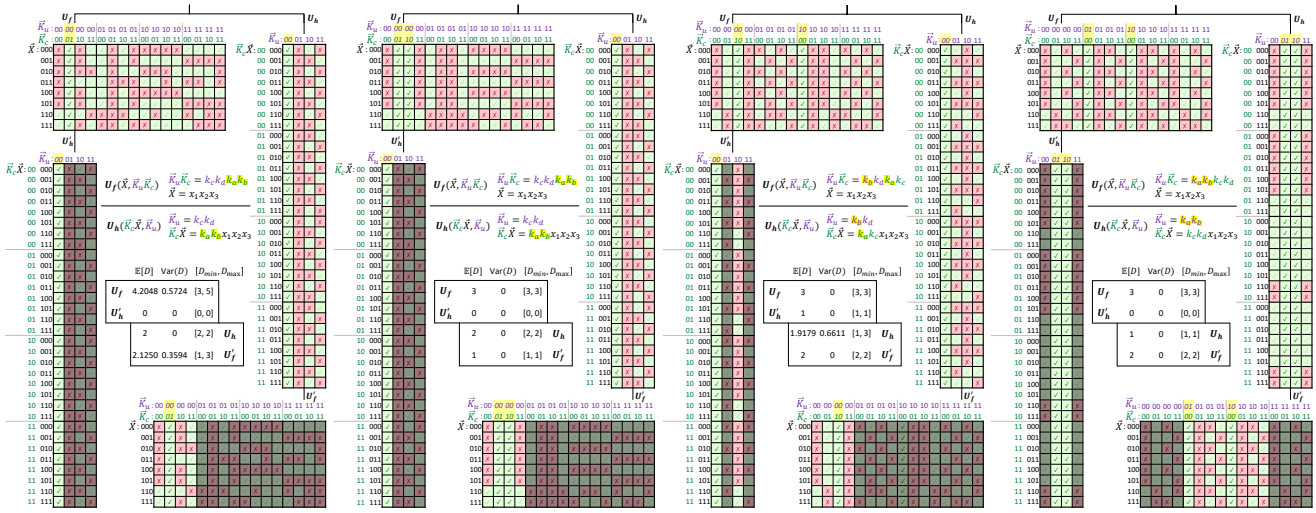


Figure 7: Original circuit and four example logic restricted circuits that demonstrate the effect of multiple correct key patterns and the controllability of the key signals with multiple correct values.



(a) Ex 1: unique $\vec{K}_u^*$ unique $\vec{K}_c^*$. **(b) Ex 2: unique $\vec{K}_u^*$ multiple $\vec{K}_c^*$.** **(c) Ex 3: multiple $\vec{K}_u^*$ multiple $\vec{K}_c^*$.** **(d) Ex 4: multiple $\vec{K}_u^*$ unique $\vec{K}_c^*$.**

Figure 8: Corresponding deobfuscation subproblems resulting from 2-stage execution of the SAT-attack on each of the four example circuits shown in Fig. 7.

the appended constraint. The active search space is the set of table entries that are not shaded in black. The smaller search space U_h' requires at most the same number of DIPs to solve as that of U_h.

The second approach to reduce the search effort is to utilize both key constraints on both rows and columns of the U_h space, where the row constraint is to keep only the row blocks with $\vec{K}_c$ pattern in the set $S_{\vec{K}_c^*}$. The additional use of the constraint on the rows does not always improve the $\mathbb{E}[D]$ in some search spaces as the removed subspace contains rows that eliminate incorrect columns more efficiently.

The third approach is to apply the joint constraint $\vec{K}_u^*\vec{K}_c^*$. The approach potentially requires fewer DIPs, but requires modification of the terminating condition of the SAT-attack. The new condition is to check for early conclusion after every determined DIP, which implies a check for the existence of a subspace (block of rows) for a correct $\vec{K}_c$ pattern where no additional DIPs are found. The approach does not directly follow the standard SAT-attack search path, and requires modifications to the analytical resilience measurement. In this paper, only the column-constraint is utilized.

For the U_h-then-U_f' approach, the key constraint for U_f' is less complex to apply, where the column subspace (block of columns) is kept if the column pattern includes $\vec{K}_u$ within $S_{\vec{K}_u^*}$, which is determined from solving U_h through execution of the first SAT-attack. After substituting $\vec{K}_u^*$, the number of clauses and variables inputted to the SAT formula expression are reduced, which results in an improved runtime.

To demonstrate the U_h-then-U_f' approach, the resulting U_h' and U_f' spaces of circuits 1 through 4 shown in Fig. 7, are provided in Fig. 8. For the four circuits shown, when the early termination of the deobfuscation methodology is considered after solving for U_f, a U_h' of $\mathbb{E}[D] = 0$ results, which implies no DIPs are required and the second SAT-attack is not needed. Although, for the example circuits provided, the transfer of the column-only constraint to the U_h' space is captured and results in early termination of the SAT-attack, that is not always the case. Some search spaces, such as the one represented by $S_{(\vec{K}_u^*,\vec{K}_c^*)} = \{(00,00),(01,00),(01,01),(10,01)\}$, require a continuous check of the early termination condition after each DIP.

4.4 SAT-resilience measure of a logic restricted circuit

If the runtime is sampled experimentally, the measure of SAT-resilience is given as the total runtime to execute the SAT-attack, which consists of the runtime of the two stages of the SAT-attack and the runtime for simplification, constraint transfer, and condition checking for early-termination of the attack. The runtime of the 2-stage SAT-attack is approximated as the sum of the runtime of each stage of the SAT-attack, which equals $(\lambda_f \times \bar{t}_f) + (\lambda_h \times \bar{t}_h)$,

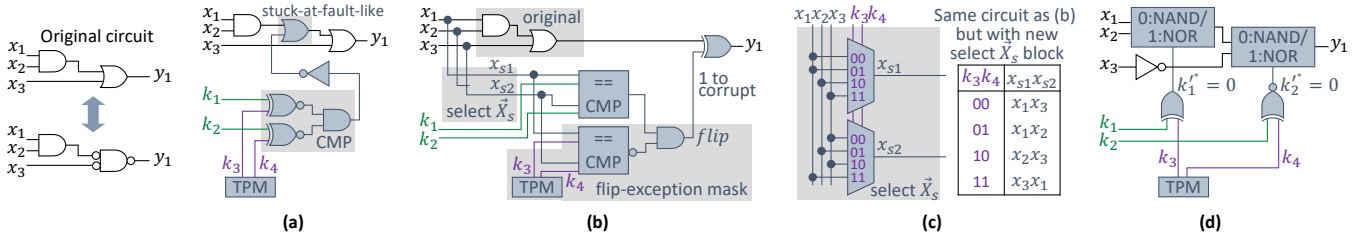


Figure 9: Logic restricted circuits that apply (a) a comparator-stuck-at-0/1-based scheme, (b) a SARLock-based scheme with $\vec{X}_s = x_1x_3$ constant, (c) a SARLock-based scheme with $\vec{X}_s$ depending on $\vec{K}_u$, and (d) an in-cone redundant key coupling scheme.

where the subscript denotes the corresponding search space. In this paper, the analytically measured total resilience is reported as two sets of scaling factors given by $(\lambda_f, \lambda_{h'})$ and $(\lambda_h, \lambda_{f'})$ for the U_f-then-U_h and U_h-then-U_f approaches, respectively. Specifically, each λ is reported as the expected resilience $\mathbb{E}[D]$. The average runtime is not reported since in this paper the number of iterations is evaluated analytically when assuming equal branching probabilities [24], without experimentally sampling SAT-attack runtimes.

5 COMPARATIVE CASE STUDY OF LOGIC RESTRICTION SCHEMES

To demonstrate the effectiveness of the proposed 2-stage SAT-attack in evaluating the SAT-resilience of a logic restricted circuit, two logic restriction techniques are compared as case studies. The two techniques implemented are comparator-based and SARLock-based, as shown in Fig. 9. The third scheme, in-cone-based using redundant key-coupling is also shown in Fig. 9 to account for other possible security techniques; however, the technique is not further discussed in the paper as an analysis similar to the other schemes applies. All three schemes utilize properties that include (1) the existence of multiple correct key patterns, (2) a maximum number of correct $\vec{K}_u$ patterns, and (3) a one-to-one relationship between $\vec{K}_c^*$ and $\vec{K}_u^*$. The three properties were first observed in the search spaces of circuits restricted by the comparator-based technique such as in papers that explored restriction of the scan chain [20, 21]. The benefit is that each fabricated restricted IC has a unique individual key value $\vec{K}_u^*$ loaded to the TPM, where the corresponding $\vec{K}_c^*$ has to be chosen correctly to activate ('de-restrict') the circuit.

As both $S_{\vec{K}_u^*} = \{0,1\}^{|\vec{K}_u|}$ and $S_{\vec{K}_c^*} = \{0,1\}^{|\vec{K}_c|}$ hold true, no columns of U_h' are eliminated when transferring constraints after solving for U_f in the first stage SAT-attack, as shown in Fig. 10. Depending on the (✓/✗) pattern in the search space, the U_h first approach tends to require at most an equal number of DIPs as compared to the U_f first approach, and is, therefore, more effective. Note that for all examples, the correct $\vec{K}_u^*$ set by the defender is 10.

5.1 Comparator-based restriction scheme

Following the comparator-based stuck-at-0/1 scheme used in [20, 21], the same mechanism is applied to a combinational circuit. Intuitively, the comparator-based scheme requires the attacker to try $\vec{K}_c$ one pattern at time, until the correct pattern is found. When the correct $\vec{K}_c$ is applied, the output drastically differs from previous attempts where an incorrect $\vec{K}_c$ pattern is used. The trial-and-error process is captured as a (✓/✗) pattern within the U_h space, as shown in Fig. 10(a). When the correct $\vec{K}_c$ is applied as part of the DIP, for

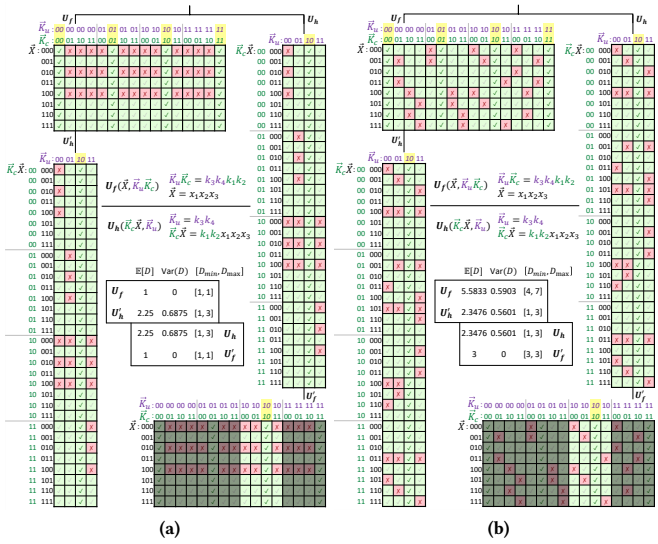


Figure 10: Corresponding U_f and U_h search spaces from execution of the 2-stage SAT-attack on the circuits shown in Fig. 9: (a) comparator-based, (b) SARLock-based with $\vec{X}_s$ depending on $\vec{K}_u$.

example the row $(\vec{K}_c, \vec{X}) = (10, 000)$, all incorrect columns of U_h are eliminated simultaneously. Note that the (✓/✗) pattern in the U_h space is similar to those of circuits locked by the TTLock [5] and SFLL-HD^0 [6] techniques.

As the comparator subcircuit depends only on $\vec{K}_c$ while being independent of $\vec{X}$, and the key signals converge to a single node (the comparator output), there are only two unique column patterns produced in the U_f space. Consequently, the U_f space is weak, which implies that comparator-based schemes are more vulnerable to attack as the security resilience is based solely on the U_h search space. However, the U_f space potentially provides high output corruption depending on the original netlist and the selected point of insertion of the "stuck-at-fault"-like gates. Note that the 1's and 0's pattern in the original circuit do not affect the SAT-resilience ($\mathbb{E}[D_f] = 1$) as long as f_o is not a constant function of 0 or 1.

5.2 SARLock-based restriction scheme

The scheme is based on the SARLock technique described in [3], where the subcircuit of the flip-exception mask is implemented using key gates instead of unobfuscated subcircuits like that in the original paper. There are two variants of the scheme as shown in Fig. 9. For the scheme shown in Fig. 9(b), the selected primary inputs

$\vec{X}_s$ are independent of $\vec{K}_u$, whereas in Fig. 9(c), $\vec{X}_s$ is configured by $\vec{K}_u$. The U_f of the SARLock-based technique shown in Fig. 9(c) is stronger than the U_f of the comparator-based technique shown in Fig. 9(a), which indicates that the U_f search space of the circuit shown in Fig. 9(a) is underutilized. The U_f' is weaker than the U_f for the circuit shown in Fig. 9(c), which indicates a benefit of transferring the constraints on the key from solving U_h first. Since the output is corrupted by XOR-ing with the flip signal, the truth-table of the original circuit does not change the U-space patterns. Therefore, the SAT-resilience and output corruptibility are independent of f_o.

6 CONCLUSION

Logic restriction is defined for the first time in contrast to other types of logic based hardware defense techniques. A novel type of deobfuscation problem emerges based on an end user attack of a logic restricted circuit that includes a combination of controllable and uncontrollable key bits. Consequently, a framework based on execution of a SAT-attack in two stages is formulated to solve the novel deobfuscation problem. An analytical measure of SAT-resilience based on work described in [24] is used along with the proposed 2-stage SAT-attack to evaluate logic restricted combinational circuits, with the characterization of the SAT resilience performed on two different restriction schemes. A visual comparison of the resulting U_f and U_h search spaces between the two different logic restriction schemes is performed, which provides insight on the strengths and weaknesses of each technique. From the case studies, for an attacker, solving for U_h and then U_f' tends to require less computational effort, especially when the defender utilizes all possible values of $\vec{K}_u^*$ with a 1:1 mapping to $\vec{K}_c^*$. Unlike logic obfuscation, where a smaller number (optimally one) of unique correct key patterns are preferred, in logic restriction, multiple correct key patterns are beneficial by enforcing extra SAT effort when executing the second stage of the attack. In addition, unlike a logic obfuscated circuit where the maximum number of DIPs possible for any search space is bounded to $\#rows = 2^{|\vec{X}|}$, the number of rows in the U_h space of a logic restricted circuit, given by $2^{|\vec{K}_c\vec{X}|}$, is unbounded as the defender is free to augment $\vec{K}_c$ with any number of additional bits at the cost of increased area and power. As a result, a logic restricted circuit is easier to secure as the U_h space provides greater SAT resilience with a larger $\vec{K}_c$.

7 ACKNOWLEDGEMENT

This work was supported in part by the Air Force Research Laboratory under Contract FA8075-14-D0025/DSTAT-15-1196, and in part by the National Science Foundation under Grant CNS-1751032.

REFERENCES

[1] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending Piracy of Integrated Circuits," *Proceedings of the IEEE/ACM Design, Automation & Test in Europe Conference*, pp. 1069–1074, Mar. 2008.

[2] K. Juretus and I. Savidis, "Reduced Overhead Gate Level Logic Encryption," *Proceedings of the IEEE/ACM International Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 15–20, May 2016.

[3] M. Yasin, B. Mazumdar, J. J. V. Rajendran, and O. Sinanoglu, "SARLock: SAT Attack Resistant Logic Locking," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 236–241, May 2016.

[4] Y. Xie and A. Srivastava, "Anti-SAT: Mitigating SAT Attack on Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 38, No. 2, pp. 199–207, Feb. 2019.

[5] M. Yasin, A. Sengupta, B. C. Schafer, Y. Makris, O. Sinanoglu, and J. J. Rajendran, "What to Lock?: Functional and Parametric Locking," *Proceedings of the IEEE/ACM Great Lakes Symposium on VLSI*, pp. 351–356, May 2017.

[6] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, "Provably-Secure Logic Locking: From Theory To Practice," *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1601–1618, Oct. 2017.

[7] R. S. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, No. 10, pp. 1493–1502, Oct. 2009.

[8] K. Juretus and I. Savidis, "Synthesis of Hidden State Transitions for Sequential Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 40, No. 1, pp. 11–23, Jan. 2021.

[9] S. Roshanisefat, H. M. Kamali, K. Zamiri Azar, S. M. Pudukotai Dinakarrao, N. Karimi, H. Homayoun, and A. Sasan, "DFSSD: Deep Faults and Shallow State Duality, A Provably Strong Obfuscation Solution for Circuits with Restricted Access to Scan Chain," *Proceedings of the IEEE VLSI Test Symposium (VTS)*, pp. 1–6, Apr. 2020.

[10] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, Vol. 8, No. 2, pp. 517–532, Aug. 2020.

[11] D. Sirone and P. Subramanyan, "Functional Analysis Attacks on Logic Locking," *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 2514–2527, Jan. 2020.

[12] X. Xu, B. Shakya, M. M. Tehranipoor, and D. Forte, "Novel Bypass Attack and BDD-Based Tradeoff Analysis Against All Known Logic Locking Attacks," *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 189–210, Sept. 2017.

[13] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "AppSAT: Approximately Deobfuscating Integrated Circuits," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 95–100, May 2017.

[14] Y. Shen and H. Zhou, "Double DIP: Re-Evaluating Security of Logic Encryption Algorithms," *Proceedings of the IEEE/ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 179–184, May 2017.

[15] M. El Massad, S. Garg, and M. Tripunitara, "Integrated Circuit (IC) Decamouflaging: Reverse Engineering Camouflaged ICs within Minutes," *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 1–14, Feb. 2015.

[16] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 137–143, May 2015.

[17] M. El Massad, S. Garg, and M. Tripunitara, "Reverse Engineering Camouflaged Sequential Circuits Without Scan Access," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 33–40, Nov. 2017.

[18] K. Shamsi, M. Li, D. Z. Pan, and Y. Jin, "KC2: Key-Condition Crunching for Fast Sequential Circuit Deobfuscation," *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 534–539, Mar. 2019.

[19] R. Karmakar, S. Chattopadhyay, and R. Kapur, "A Scan Obfuscation Guided Design-for-Security Approach for Sequential Circuits," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 67, No. 3, pp. 546–550, Mar. 2020.

[20] Q. L. Nguyen, E. Valea, M. L. Flottes, S. Dupuis, and B. Rouzeyre, "A Secure Scan Controller for Protecting Logic Locking," *2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 1–6, July 2020.

[21] R. Karmakar and S. Chattopadhyay, "On Securing Scan Obfuscation Strategies Against ScanSAT Attack," *Proceedings of the IEEE International Symposium on Quality Electronic Design (ISQED)*, pp. 213–218, Mar. 2020.

[22] K. Juretus and I. Savidis, "Reducing Logic Locking Key Leakage through the Scan Chain," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, Oct. 2020.

[23] L. Alrahis, M. Yasin, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "ScanSAT: Unlocking Obfuscated Scan Chains," *Proceedings of the Asia and South Pacific Design Automation Conference*, p. 352–357, Jan. 2019.

[24] S. Phatharodom, N. Kandasamy, and I. Savidis, "Modeling SAT-Attack Search Complexity," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, Oct. 2020.

[25] K. Juretus and I. Savidis, "Importance of Multi-parameter SAT Attack Exploration for Integrated Circuit Security," *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 366–369, Oct. 2018.

[26] P. Tuyls, G. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-Proof Hardware from Protective Coatings," *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 369–383, Oct. 2006.

[27] M. Yasin and O. Sinanoglu, "Transforming Between Logic Locking and IC Camouflaging," *Proceedings of the IEEE International Design for Test Symposium (IDT)*, pp. 1–4, Dec. 2015.

[28] Y. Hu, V. V. Menon, A. Schmidt, J. Monson, M. French, and P. Nuzzo, "Security-Driven Metrics and Models for Efficient Evaluation of Logic Encryption Schemes," *Proceedings of the 17th ACM-IEEE International Conference on Formal Methods and Models for System Design*, pp. 1–5, Oct. 2019.