A Secure Scan Controller for Protecting Logic Locking

Quang-Linh Nguyen, Emanuele Valea, Marie-Lise Flottes, Sophie Dupuis, Bruno Rouzeyre

A Secure Scan Controller for Protecting Logic Locking

Quang-Linh Nguyen, Emanuele Valea, Marie-Lise Flottes, Sophie Dupuis, Bruno Rouzeyre
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
firstname.lastname@lirmm.fr

***Abstract*—The globalized supply chain in the Integrated Circuit (IC) industry raises several security concerns such as overproduction, IP piracy and Hardware Trojan insertion. Logic locking has emerged as a potential countermeasure to address these issues. However, its efficiency is challenged by various attacks, especially oracle-guided attacks based on Boolean Satisfiability (SAT) solvers. These attacks rely on the possibility for an attacker to control and observe in the field the internal state of a functional IC, which acts as an oracle. This ability to control/observe the IC states is offered by scan chains, typically used for IC production testing. In this paper, we propose a method, complementary to logic locking, to prevent such attacks. This method introduces a scan chain controller with a key-based authentication mechanism, in order to prevent unauthorized access to the scan chains once the IC is deployed in the field. The solution can be coupled with any logic locking technique at the cost of negligible area overhead. Furthermore, it is secure against state-of-the-art attacks and supports full testing.**

***Index Terms*—IP Piracy, Design-for-Security, Logic Locking, Design-for-Trust, Scan Chains**

I. Introduction

Intellectual property (IP) infringement is a well recognized problem in the semiconductor industry. This issue stems from the globalization of the supply chain. Ever-shrinking technologies have rapidly raised the cost of manufacturing Integrated Circuits (ICs). Therefore, outsourcing the fabrication process to offshore foundries has become a major trend [1]. Besides, using commercial third-party IPs allows companies to reduce design effort and, hence, time-to-market. However, different actors in this segmented IC supply chain may violate agreements by overproducing ICs, inserting malicious hardware Trojans or illegally reusing out-of-contract IPs.

While former counterfeit detection techniques are based on long and expensive parametric tests performed on suspected ICs, numerous recent Design-for-Trust approaches introduce preventive mechanisms at design time: sensors detect die and IC recycling [2]; IC camouflaging prevents reverse engineering [3]; hardware watermarking is used for demonstrating compliance with the ownership [4]; passive hardware metering, based on physical unclonable functions or digitally stored serial numbers allows IC identification. Last but not least, active hardware metering consists in locking each IC until a key is provided by the IP holder. This solution can be used at IP core level to restrict illegal IP reuse, and at system level to prevent IC overproduction and, to some extent, prevent hardware Trojan insertion.

Logic locking [5]–[7] is an active hardware metering technique dedicated to combinational structures. This approach consists in modifying the original netlist with additional logic controlled by additional key inputs. The modified design behaves as the original one only upon the application of the correct key value at the key inputs; otherwise, it outputs erroneous values. Only the designer who applies logic locking to the design knows the correct key value. When the modified design is sent to outsourced services for fabrication, test and packaging, the produced ICs are "locked" to an incorrect functionality, and thus unusable, until they are sent back to the designer for activation. The ICs are "unlocked" when the designer programs the correct key value in their tamper proof memory. Traditional logic locking techniques insert key-gates such as XOR/XNOR gates or multiplexers [8]–[10]. Key-gate locations can be random or strategic, for example, to avoid key recovery through netlist analysis or to maximize corruption.

Logic locking methods have been challenged by various oracle-guided attacks aiming at recovering the secret key. The attack model requires the access to two fundamental assets:

- a *locked netlist*, i.e. the reverse-engineered netlist containing the logic-locking structure;
- an *oracle*, i.e. an unlocked IC with accessible scan chains.

Sensitization attacks [11] identify the correct key bits by analyzing logic cones and sensitizing each key bit to observable outputs. This task requires that the targeted logic cones' inputs are controllable (Primary Inputs (PIs) or scan Flip-Flops (FFs)) and their outputs are observable (Primary Outputs (POs) or scan FFs). The attack can be prevented by making the sensitization of each key bit dependent on the other key bits.

Oracle-guided attacks using the Boolean Satisfiability (SAT) solver are, by far, the most effective attacks [12]–[14]. They exploit the reasoning capability of a SAT solver to quickly eliminate wrong key values. A preliminary requirement of such attacks is to model the circuit as a Direct Acyclic Graph (DAG). However, combinational circuits can be interpreted as DAGs whereas sequential circuits cannot. A sequential circuit is modelled as a combinational circuit by converting internal FFs' outputs into pseudo-primary inputs (PPIs) and their inputs into pseudo-primary outputs (PPOs). Key values are correlated and grouped into equivalence classes. In each iteration, the attack chooses two key values from two classes and finds a so-

called Distinguish Input Pattern (DIP) that results in different output values for the two key values when applied to the locked netlist. This DIP is then applied to the oracle to obtain the correct output. By comparing the different outputs, the attack is able to suppress at least one class of key values, which may contain multiple values. The key search space is reduced iteratively until the SAT solver deduces the correct key.

Existing countermeasures against SAT-based attacks focus mainly on increasing the number of iterations of the attack [15], [16]. However, such solutions provide very low output corruption. For each wrong key value, only one (or very few) input pattern results in incorrect output. Furthermore, the additional locking structure is greatly distinguishable from the original netlist and, thus, can be easily removed [17].

As mentioned above, most attacks on logic locking are based on the opportunity to control the inputs and to observe the outputs of the combinational block under attack. Because combinational blocks are generally surrounded by sequential elements (i.e. FFs) for synchronisation, all previous works assume that internal FFs of the unlocked IC can be accessed. This is possible since original FFs are generally replaced by scan FFs for production test purpose. Scan design is indeed a fundamental Design-for-Testability approach [18] where scan FFs are linked together to form shift registers, so-called *scan chains*. A scan chain's input is a fully controllable PI and its output is a fully observable PO. By running shift operations in the scan chain, data stored into scan FFs can be read and modified. In addition, scan FF outputs and inputs are considered as the combinational part's PPIs and PPOs respectively. This allows test generation tools to model sequential designs as combinational, which decreases the test generation complexity. As the access to scan chains satisfies the model requirement of mentioned attacks on logic locking, they can be prevented by including an authentication step to block scan shift operations performed by the attackers.

In this paper, we present a scan solution that guarantees full testability without compromising the security of logic locking. We propose a scan controller that limits the scan access only to authorized users. We provide suitable logic locking techniques to combine with the proposed solution. A security analysis shows its capability to prevent SAT-based attacks on logic locking. Furthermore, we analyse potential attack schemes on the scan controller.

The rest of the paper is organized as follows. Section II details the threat model and the security requirements. Section III presents the proposed scan protection technique. The analysis of the solution is detailed in Section IV. Section V shows a comparison of our proposal with related works. Finally, Section VI concludes the paper.

II. Threat Model

Designers can protect their hardware IPs by applying logic locking and our proposed scan controller. In this case, the design is locked with a *logic locking key*, referred to as K_L; and its scan chains are locked with a *scan access key*, referred to as K_S.

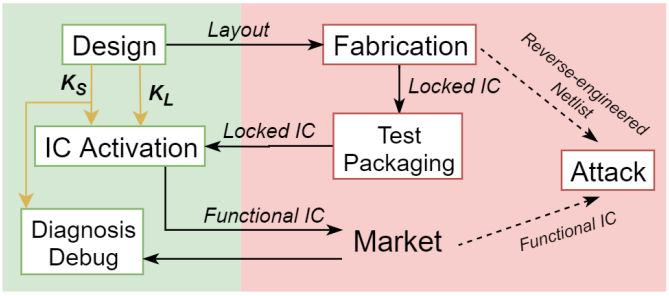

Fig. 1. Production flow and attack model.

Fig. 1 depicts the threat model corresponding to this scheme. The foundry, the test/packaging facility and the end user, depicted on a red background in Fig. 1, are potential attackers who want to recover K_L. After fabrication and packaging, the ICs are sent to the designer or a trusted entity, who programs, into their tamper-proof memory (TPM), K_L for unlocking their functionality and K_S for securing the scan chains. Thus, the unlocked ICs have "locked" scan chains controlled by the scan controller. According to the model of the SAT attack and the sensitization attack, the following assets are required: (i) a locked netlist; (ii) an unlocked IC; (iii) K_S.

The attacker's capabilities are defined as follows. We assume that unlocked ICs can be purchased from the market and, hence, accessible to all potential attackers. The manufacturer receives the layout of the logic locked design for fabrication. The locked netlist can be obtained by reverse-engineering the layout. The end user or the test facility can obtain the locked netlist by trading with the manufacturer or by reverse-engineering an unlocked circuit if they have a capable facility.

According to the threat model, the attackers are not in possession of K_S. The designer should restrict the distribution of K_S only to trusted partners. The manufacturer is commonly in charge of conducting production test. It includes a scan-based structural test that is performed on wafers by probing each die. For this structural test, test patterns are generated with incorrect values of K_L. Hence, K_L is not required for performing production test. However, using scan chains is necessary. Thus the scan controller should be temporarily bypassed during this phase. Since K_S is also required for further failure analysis when an IC presents an erroneous behavior in the field, debug and diagnosis must be conducted by the design house or a trusted partner.

III. Proposed Solution

The main idea of the proposed solution is to use a key-based authentication for controlling the activation of scan chains. Upon the insertion of an incorrect key value to the scan controller, no scan data is available for attacking purposes. To further increase the robustness of the scheme, the scan

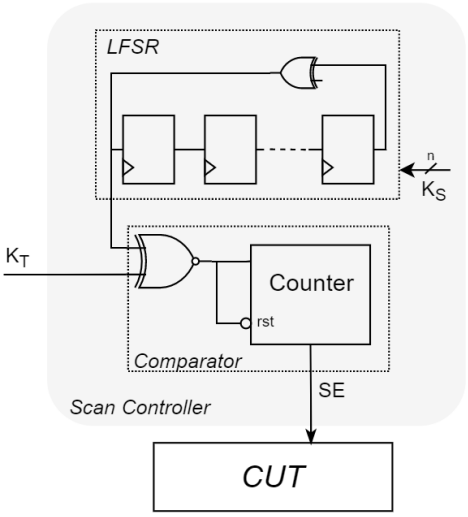

Fig. 2. Structure of the proposed scan controller.

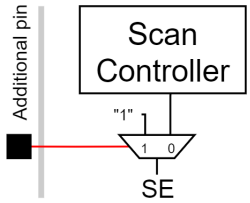

Fig. 3. Bypassing the scan controller during manufacturing test.

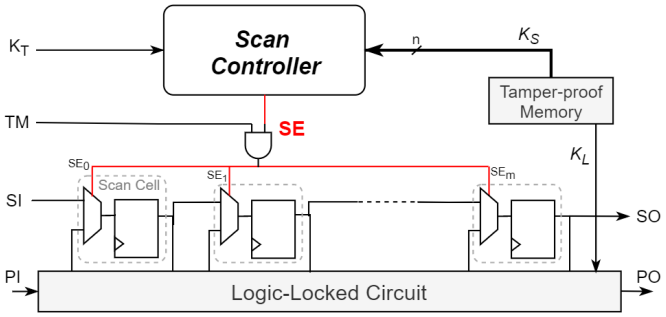

Fig. 4. Using the scan controller with a logic-locked circuit.

controller includes a linear-feedback shift register (LFSR) that dynamically changes the value of the required key.

A. Scan Controller

The structure of the proposed scan chain controller is illustrated in Fig. 2. It includes an n-bit linear-feedback shift register and a comparator. The scan controller "locks" the Scan Enable (SE) signal by setting it to "0". SE is set to "1" only upon the insertion of the correct dynamic value of *test key*, referred to as K_T, that matches the output of the LFSR. K_S initializes the LFSR and, thus, defines the correct K_T value.

K_S is stored in a TPM that is programmed by the designer during the activation phase as depicted in Fig. 1. It is possible to program each IC with a different value of K_S. Therefore, each IC requires a distinct K_T bit stream that can not be reused on another IC. This property can be exploited to help designers track scan usage in each IC.

The LFSR is an n-bit shift register. It randomizes its state at every clock cycle, i.e. $St_{LFSR_t}(K_S) \neq St_{LFSR_{t+1}}(K_S)$ at any cycle t. As the LFSR is used as a bit stream generator, its output throughout n cycles represents one of its states. The comparator consists of a XNOR gate and a counter. The XNOR gate compares the inserted K_T value and the current output of the LFSR, and feeds the result to the counter. The counter checks if the K_T stream matches the state of the LFSR, i.e. the value of K_T is correct for n *consecutive* cycles. If "1" has been inputted to the counter for the last n cycles, it sets SE signal to "1"; otherwise, SE remains at "0". If the comparison result becomes "0", the counter resets immediately and SE is set to "0". Given the correct K_T bit stream, after the first $n-1$ cycles, SE is set to "1" continuously.

To switch scan cells from functional mode to shift mode, a Test Mode (TM) signal is used to set the SE pin in each scan cell to "1". The TM signal also avoids unwanted mode switching due to the periodic nature of the LFSR.

The scan controller should be temporarily bypassed during manufacturing test to avoid communicating K_S to adversaries. This can be done by physically connecting the output of the scan controller to a pull-up element or a controllable source. The former is based on the Saw Bow method [19], where the connecting wire crosses the sawing line of the wafer. The latter is depicted in Fig. 3. An additional pin directly controls the value of the SE signal. It is blown off after the packaging process.

Upon knowledge of K_S (i.e., its seed) and its structure, the LFSR can be recreated. An authorized tester who received the mentioned assets can build an equivalent model of the LFSR to generate the required K_T bit stream.

B. Combining with Logic Locking

The proposed scan solution can be combined with any logic locking technique, as shown in Fig. 4, to increase the resilience against attacks that rely on scan chain access. Any related state-of-the-art logic locking technique focusing on identifying strong key-gate locations [9] and/or maximizing output corruption [8], [10] can be used to complement the scan shift protection mechanism provided by our solution.

IV. Analysis

A. Security Analysis

1) Protecting Logic Locking: The SAT attack needs an oracle with accessible scan chains to control and observe respectively inputs and outputs of the combinational part under attack. The proposed scan controller blocks any unauthorized usage of the scan chains in the oracle. In other words, the attacker cannot apply any DIP generated by the SAT solver to the inputs of the combinational part, and he/she cannot

deduce expected outputs for wrong key elimination. Therefore, the attack is unable to solve the problem of finding K_L. The same remarks can be made concerning the sensitization attacks, which also require full control and observation of combinational logic cones.

A more computationally intensive way to represent a sequential circuit without scan chains as combinational is to use the time-frame expansion method. The circuit is modelled as a series of copies of its combinational part, where each copy corresponds to a time-frame. The same procedure is used to perform production test pattern generation on sequential circuits without scan chains. However, the number of time-frames must be large enough to reach all the states of the circuit, which can require up to $2^{n_{FF}}$ time-frames, where n_{FF} is the number of FFs. For this reason, sequential test pattern generation is avoided in practice and scan design is the *de facto* Design-for-Testability approach today. In [20], [21], authors report that attacks based on "sequential" SAT approaches can only handle small circuits with a few thousands of gates.

As mentioned in Section III-B, key-gate insertion locking techniques should be used with the scan controller. After synthesis, inserted key-gates and their neighboring gates are structurally transformed into different forms, making them resilient against removal. Furthermore, these techniques have higher output corruption and smaller overhead compared to state-of-the-art SAT-resilient techniques.

2) Securing the Scan Access Key: K_S needs to be kept secret; otherwise, an attacker with the reverse-engineered netlist can recreate the LFSR inside the scan controller to generate K_T and activate the scan chains. However, the nature of the scan controller gives minimal output observability to attackers:

- SE is the only output signal of the scan controller and it is just 1-bit wide.
- SE is an internal signal of the circuit, which makes observing its behavior challenging for attackers.

As mentioned in section III-A, K_T has to be correct for the last n cycles in order to enable the scan chains for 1 cycle. The probability of guessing this value with a brute-force approach is $1/2^n$. The function of the scan controller at any cycle t can be modelled as a point function:

$$\begin{cases} 1, & \text{if } K_{T\,t\rightarrow t+n} = St_{LFSR_t}(K_S) \\ 0, & \text{otherwise} \end{cases} \quad \forall K_S, K_{T\,t\rightarrow t+n} \in \{0,1\}^n \tag{1}$$

A point function is a one-way function, which is a fundamental cryptographic primitive and is provably hard to invert. An attacker can resort to cryptanalytic principles to correlate the inserted K_T and the SE signal, observed through scan output. However, due to the point function behavior, it is not possible to derive K_S with a more efficient approach than brute-force guessing.

Point functions have also been proven to be resilient against SAT-based attacks [15], [16]. Using a point function to hide secret information forces the SAT solver to check for every possibility. In other words, the SAT solver is forced to generate a new DIP for the elimination of only one key at every iteration, hence reducing the attack's efficiency to that of brute-force. Therefore, using formal methods cannot aid attackers at retrieving the K_S value.

Attackers who own the reverse-engineered netlist can resort to ScanSAT attack [22]. Based on a SAT solver, it targets scan protection methods that corrupt scan data in the absence of a secret key. Using the same attack model as the SAT attack on logic locking, ScanSAT depends on scan data control and observation. However, as mentioned in Section III, data from the scan chains are not available since the scan controller disables the scan chains. Furthermore, data observed at SO pin show no correlation with K_S. Thus, ScanSAT is not effective against our scan protection.

The scan controller also improves data confidentiality in the circuit. Scan chains are the target of a plethora of attacks that aim to steal secret data from ICs [23]. The scan controller prevents attackers from using scan structure deliberately. Thus, malicious data cannot enter the scan chains, nor can secret data be examined from the scan chains.

3) Tampering: An untrusted manufacturer has the capability to modify the mask before IC fabrication, which allows them to bypass the scan controller introduced in the original design. However, this modification is easily detectable since scan shift would be allowed without the need to provide the correct K_T. We assume that an untrusted packaging facility neglects to destroy the additional pin used to bypass the scan controller during production test. As the produced ICs are sent back to the design house for activation, the designers can perform an *ad hoc* test to detect such tampering before inserting K_L.

B. Testability Analysis

Any fault in the scan controller easily propagates to the SE signal and affects the activation of the scan chains. Thus, the scan controller can be tested with a functional test. Furthermore, the built-in LFSR acts as a self-test structure to facilitate the test.

Structural test can be applied to the CUT before it is unlocked. Test patterns for such test are generated assuming that key inputs are controllable; thus, the correct K_L value is not required. This procedure is shown to provide maximum fault coverage while preventing attackers from retrieving K_L by analysing test patterns [24].

C. Overhead

We implemented the scan controller in ITC'99 benchmarks [27]. The benchmarks were synthesized on a 65nm technology library with Synopsis Design Compiler [28]. Fig. 5 shows the area overhead for three versions of the proposed scan controller using a 64-bit, an 80-bit and a 128-bit LFSR, along with a comparison with related works [25], [26] using a 128-bit key. The area overhead of the scan controller is as low as 0.3% in the b19 benchmark. Although the solution in [26] has

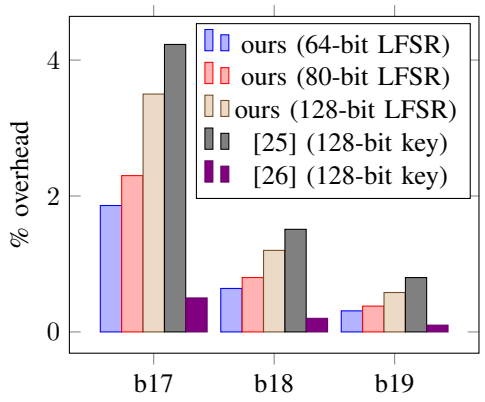

Fig. 5. Area overhead of the proposed scan controller and related works on ITC'99 benchmarks.

a smaller area cost, we show in the next section that it is less secure than the proposed solution.

We used Synopsis TetraMax [29] to generate test patterns for the mentioned benchmarks. The b18 benchmark is estimated to have a test time of about $6 \cdot 10^7$ clock cycles. Using a 128-bit scan controller takes 127 initial cycles to set up the SE signal. Thus, it represents a largely trivial test time overhead.

V. Comparison with Related Works

Research on secure scan chain has been established since the introduction of scan-based side channel attacks on crypto circuits [23]. The attacks deduce the cipher key by analyzing the partial results of the cryptographic operation that can be observed through the scan chains. However, countermeasures against such attacks may not adapt to the threat model of attacks on logic locking, where the attacker has the additional capability of reverse-engineering the circuit. Solutions that aim at masking scan data and assume that the scan chain structure is unknown to the attacker, such as [30] which judiciously inserts inverters into the scan chains, are therefore vulnerable in the scenario of logic locking. Other solutions based on manipulating scan data during the switch between functional mode and test mode [31] also turn nullified as SAT attack or ScanSAT attack only work in test mode. Therefore, recent works have proposed secure scan solutions that accommodate the stronger threat of attacks on logic locking.

One class of solutions is based on scan obfuscation. Using the same principle as logic locking, Encrypt Flip-Flop technique [26] inserts key-gates in the scan chains to corrupt scan data, both at shift in and shift out operations. A stronger obfuscation effect can be achieved when the secret key changes dynamically, which makes the function of scan-data transformation change over time. Such dynamic scan obfuscation strategy is presented in [32]. An LFSR is connected to the key inputs of key-gates and it feeds new key values periodically. However, as is the case with logic locking, the obfuscated scan output reflects the inversion effect of key-gates and it can be used to trace back to the secret key. ScanSAT attack is indeed capable of modelling the locked scan chains as a combination of key-gates inserted at PPIs and PPOs. Using the same attack model as SAT attack, the secret key is deduced by the SAT solver.

TABLE I. Comparison with other scan-based SAT-resistant techniques

Solution	Attacks	
	ScanSAT [22]	Shift & Leak [33]
Encrypt Flip-Flop [26]	×	✓
Dynamic Scan Obfuscation [32]	×	✓
Secure Cell [25]	✓	×
Proposed	✓	✓

✓ denotes resilience against the attack
× denotes susceptibility to the attack

Guin et al. [25] proposed a solution to counteract the SAT attack by restricting the observation of scan output. The logic-locking key is hold inside modified scan cells, called Secure Cells (SC). A SC holds its previous state at test time. In addition, a test suppressor is introduced to delete scan output data whenever the circuit switches from functional mode to test mode, hence, preventing leaking the key through scan data. However, due to testing support, the solution does not control the scan shift-in operation, which is exploited by a customized attack called shift-and-leak [33]. The key bits can be shifted to easily observable scan cells and test patterns can be used to set a condition for sensitizing each key bit to the POs.

Our proposed scan controller is shown to be resistant to ScanSAT (section IV-A2) while demanding minimal area cost. A comparison between the proposal and the mentioned solutions is presented in Table I.

VI. Conclusion

The scalability and efficiency of SAT-based attacks, as well as sensitization-based attacks, on logic locking rely heavily on the control and observation of inputs/outputs of the combinational part under attack. While scan chains are widely used in the industry for production test purpose, they also provide such capability and, thus, make these attacks possible.

In this paper, we present a low-cost scan controller that requires a dynamic key to enable the shift operations in the scan chains. The resulting blocked scan access prevents adversaries from shifting data in or out from the scan chains, making it impossible to implement the attacks. Using the scan controller with key-gate insertion techniques meets all the requirements of logic locking. Security analysis shows its ability to secure the scan access key and its robustness against tampering. Full testing is supported when performed by authorized partners without leaking secret data to adversaries. In addition, the solution is scalable and easy to integrate.

Acknowledgment

This work is funded by project MOOSIC ANR-18-CE39-0005 of the French National Research Agency (ANR).

References

[1] R. Kumar, "Simply Fabless!," *IEEE Solid-State Circuits Magazine*, vol. 3, no. 4, pp. 8–14, 2011.

[2] X. Zhang and M. Tehranipoor, "Design of On-Chip Lightweight Sensors for Effective Detection of Recycled ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, pp. 1016–1029, May 2014.

[3] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13*, (Berlin, Germany), pp. 709–720, ACM Press, 2013.

[4] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking techniques for intellectual property protection," in *Proceedings of the 35th Annual Conference on Design Automation Conference - DAC '98*, (San Francisco, California, United States), pp. 776–781, ACM Press, 1998.

[5] S. Dupuis and M.-L. Flottes, "Logic Locking: A Survey of Proposed Methods and Evaluation Metrics," *J Electron Test*, May 2019.

[6] A. Chakraborty, N. G. Jayasankaran, Y. Liu, J. Rajendran, O. Sinanoglu, A. Srivastava, Y. Xie, M. Yasin, and M. Zuzak, "Keynote: A Disquisition on Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[7] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP Protection and Supply Chain Security Through Logic Obfuscation: A Systematic Overview," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 24, pp. 65:1–65:36, Sept. 2019.

[8] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," *IEEE Transactions on Computers*, vol. 64, pp. 410–424, Feb. 2015.

[9] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 1411–1424, Sept. 2016.

[10] R. Karmakar, H. Kumar, and S. Chattopadhyay, "On Finding Suitable Key-Gate Locations In Logic Encryption," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2018.

[11] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Design Automation Conference*, pp. 83–89, June 2012.

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

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

[14] K. Z. Azar, H. M. Kamali, H. Homayoun, and A. Sasan, "SMT Attack: Next Generation Attack on Obfuscated Circuits with Capabilities and Performance Beyond the SAT Attacks," *1*, pp. 97–122, 2019.

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

[16] Y. Xie and A. Srivastava, "Mitigating SAT Attack on Logic Locking," in *Cryptographic Hardware and Embedded Systems – CHES 2016*, vol. 9813, pp. 127–146, 2016.

[17] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[18] L.-T. L.-T. Wang, X. Wen, and K. S. Abdel-Hafez, "Design for Testability," in *VLSI Test Principles and Architectures*, pp. 37–103, Elsevier, 2006.

[19] G. Di Natale, M.-L. Flottes, B. Rouzeyre, and P.-H. Pugliesi-Conti, "Manufacturing Testing and Security Countermeasures," in *Hardware Security and Trust: Design and Deployment of Integrated Circuits in a Threatened Environment* (N. Sklavos, R. Chaves, G. Di Natale, and F. Regazzoni, eds.), pp. 127–148, Cham: Springer International Publishing, 2017.

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

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

[22] L. Alrahis, M. Yasin, N. Limaye, H. Saleh, B. Mohammad, M. Al-Qutayri, and O. Sinanoglu, "ScanSAT: Unlocking Static and Dynamic Scan Obfuscation," *IEEE Transactions on Emerging Topics in Computing*, Sept. 2019.

[23] E. Valea, M. D. Silva, G. D. Natale, M. Flottes, and B. Rouzeyre, "A Survey on Security Threats and Countermeasures in IEEE Test Standards," *IEEE Design & Test*, vol. 36, no. 3, pp. 95–116, 2019.

[24] M. Yasin, S. M. Saeed, J. Rajendran, and O. Sinanoglu, "Activation of logic encrypted chips: Pre-test or post-test?," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 139–144, Mar. 2016.

[25] 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*, vol. 26, pp. 818–830, May 2018.

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

[27] F. Corno, M. Sonza Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design & Test of Computers*, vol. 17, pp. 44–53, July 2000.

[28] "Design Compiler®." https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/design-compiler-nxt.html. Accessed: May 19, 2020.

[29] "Tetramax." https://www.synopsys.com/support/training/signoff/tmax1-fcd.html. Accessed: May 19, 2020.

[30] 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.

[31] D. Hely, F. Bancel, M. Flottes, and B. Rouzeyre, "Test control for secure scan designs," in *European Test Symposium (ETS)*, pp. 190–195, May 2005.

[32] X. Wang, D. Zhang, M. He, D. Su, and M. Tehranipoor, "Secure Scan and Test Using Obfuscation Throughout Supply Chain," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 1867–1880, Sept. 2018.

[33] N. Limaye, A. Sengupta, M. Nabeel, and O. Sinanoglu, "Is Robust Design-for-Security Robust Enough? Attack on Locked Circuits with Restricted Scan Chain Access," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, Nov. 2019.