Modeling SAT-attack Search Complexity

Saran Phatharodom, Nagarajan Kandasamy, and Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University, Philadelphia, PA 19104
sp694@drexel.edu, kandasamy@drexel.edu, isavidis@coe.drexel.edu

***Abstract*—In this paper, a metric based on mathematical modeling is proposed to evaluate the strength in security of a logic-locked circuit against a satisfiability (SAT) based attack. Current approaches estimate the SAT resilience experimentally based on time-to-solve or the number of calls to a SAT-solver. However, the estimate is often based on one sample or a small sample size. Due to the possible variation in the search path length of the SAT-attack, a measure of resilience based on statistical characterization is proposed. A probabilistic model of a SAT-attack search process is developed to properly capture the variation in the path length and report the SAT resilience as an expectation of the computational complexity. An estimator of the expected complexity, assuming an equally likely branching probability, is proposed. The model and the estimator allow for 1) the derivation of a closed-form estimate of the expected security, and 2) characterization of the key search space without experimental bias toward SAT-attack implementation or circuit topology. As a case study, an analysis of the security gain per inserted key gate is performed on a full adder circuit. The study reveals a monotonically increasing resilience and provides insights on the most efficient key gate placement strategy that maximizes the achievable security.**

I. Introduction

As a means to counter hardware threats to the combinational logic of an integrated circuit (IC), various logic locking techniques [1]–[4] have been developed through the past decade. In response, multiple novel attacks [5]–[7] have been proposed to exploit the various weaknesses of the obfuscation techniques, of which the satisfiability (SAT) based attack [8], [9] remains one of the fundamental threats to logic locked circuits to date. However, despite the constant need for a reliable method to evaluate the proposed locking techniques, an in-depth measure of SAT-resilience is not widely discussed and is often overlooked. Typically, a logic locking technique is proven to be resilient against a SAT-attack based on experimentally sampled time-to-solve characteristics or a count of the number of iterations a SAT solver is called. Although experimental evaluation provides a sound means to measure security, current practices do not properly evaluate resilience and often undermine the reliability of the results.

1) Undersampling and need for stochastic measures: For some locked circuits, given the same SAT-attack instance, there is potential variation in the number of iterations required to solve for the key [3], [10], [11]. As a result, a deterministic implementation of a SAT-attack or an under-sampled stochastic implementation likely misrepresent the effectiveness of a locking technique. A careful configuration of a stochastic implementation is needed to properly explore the key search space.

2) Benchmarking using large key sizes: Prior studies attempt to experimentally validate the security of a proposed locking technique by demonstrating an exponential growth in the computational cost of executing a SAT-attack for a linear increase in the key size [2], [11], [12]. In addition, prior studies aim to prove a technique secure by applying a large key that results in a time-out when executing the SAT-attack for an arbitrarily set maximum run time [11], [13]. As a result, relatively large keys are selected for evaluation. Given a fixed allocated experimental run time, a fewer number of locked circuit instances are evaluated against the SAT-attack, which limits the amount of measured data points. In addition, if the SAT-attack times out, the measure of the search effort is not truly observed. The trend in the gained security as a function of the key size is, therefore, not rigorously validated.

Two novel concepts are described in the paper. First, due to the observed variable nature of the number of SAT-attack iterations, a probabilistic model of the search time complexity of a SAT-attack is developed in order to derive a statistical measure of the SAT-resilience. Second, based on the probabilistic model, a framework to *theoretically estimate* the security of an obfuscated circuit is developed, which complements current approaches that apply *experimental sample means*.

A probabilistic measure of SAT-resilience, specifically, the expected number of iterations, is used in [3]. However, the paper does not properly formulate or define the random process in detail. In this paper, the SAT-attack search process is first explored as a *minimum set cover problem*. Next, given a small locked circuit, all possible SAT-attack search paths are enumerated and visualized as a tree diagram. Assuming a given branching probability, the tree diagram forms a probability space, in which a probability mass function (PMF) of the search lengths is derived. A truth-table of the uncorrupted output of a circuit is proposed to represent the SAT-attack search process, which is independent of both the circuit topology and the functionality of the circuit.

An analysis is performed on two applications of the model while also utilizing the developed SAT-resilience statistical estimators that assume an equally-likely branching probability. The benefit of applying the theoretical model to produce closed-form statistical expressions of the resilience to SAT-attacks and to analyze the change in resilience due to the insertion of a key gate is demonstrated.

II. Modeling the SAT-attack search problem

When performing a SAT-attack, an adversary is assumed to have access to 1) an activated locked IC bought from the market and 2) an extracted netlist of the locked circuit with functionality $f(\vec{X}, \vec{K})$ for input pattern $\vec{X}$ and key vector $\vec{K}$. The netlist is extracted by either reverse engineering the locked-IC or by having access as a third-party affiliate within the supply chain. The activated locked IC (the oracle), with circuit functionality given by $f(\vec{X}, \vec{K}^{correct})$, is logically equivalent to the original unobfuscated circuit with functionality of $f_o(\vec{X})$. However, the adversary cannot directly access the correct key bits since the key is loaded into a tamper-proof memory [14]. For the adversary, the goal is to determine the correct key vector, for which the SAT-attack is an efficient algorithm that utilizes a commonly available SAT-solver to prune the key search space.

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

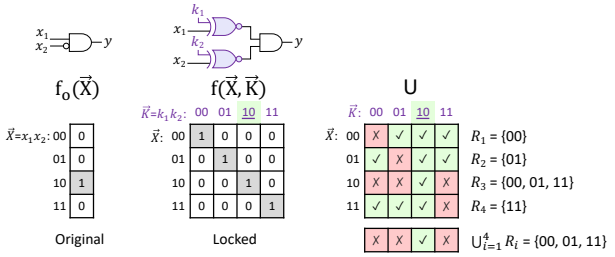

Fig. 1: A logic locked circuit f and an original circuit f_o with the resulting column elimination table, which demonstrates deobfuscation of f by applying the SAT-attack.

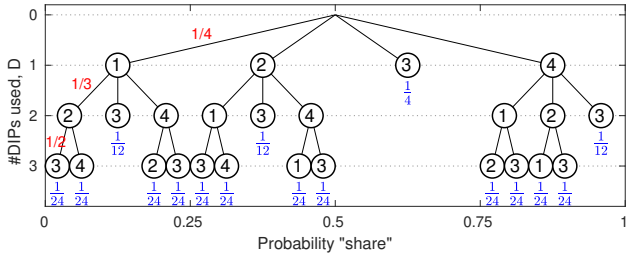

Fig. 2: A DIP-sequence tree for a SAT-attack applied to the locked circuit shown in Fig. 1. Assuming an equally likely branching probability, the probability of each branching on the DIP sequence (R_1, R_2, R_3) is shown in red, and the probability of the SAT-attack choosing a given DIP sequence is shown in blue directly below the lowest node of a given path.

A. Incorrect key pattern elimination and set cover problem

Given an activated IC and a logic locked circuit, applying a specific input pattern on both circuits and comparing the outputs leads to the elimination of a subset of incorrect key patterns. As an example, a small instance of an and-tree with locked primary-inputs, as shown in Fig. 1, is analyzed. Querying the oracle using $\vec{X} = 00$, the returned output is $f_o(00) = 0$. The input 00 eliminates the key pattern given by column 00 in Fig. 1, since $f(00, 00) \neq 0$.

Let the "row set" R_i denote the set of key patterns eliminated by the use of the input pattern of the i^{th} row. To eliminate all incorrect key patterns, a collection of row sets is needed, in which there are multiple combinations of rows that cover all incorrect key sequences including the subset $\{R_3\}$, $\{R_1, R_2, R_4\}$, and $\{R_3, R_4\}$. Determining the smallest collection of row sets is equivalent to the *minimum set cover problem*. With respect to the formulation of the set cover problem, the *universe* is given by $S_U = \cup_{i=1}^{2^{|\vec{X}|}} R_i$, which evaluates to the set of all incorrect key patterns. Let S_R denote the multiset of all row sets $\{R_1, R_2, ..., R_{2^{|\vec{X}|}}\}$. The objective of deobfuscation is to determine a *cover* set $S_C \subseteq S_R$ such that $\cup_{R \in S_C} R = S_U$.

B. SAT-attack search complexity

Abstractly, the SAT-attack algorithm executes the column elimination process iteratively by appending one row set at a time to the cover set. Each newly added row set must eliminate at least one column with an incorrect key pattern that was not previously removed. The key search process, therefore, slightly differs from the set cover problem, as the cover set is now order-sensitive. For example, $\{R_3, R_4\}$ is a valid cover set. However, while (R_4, R_3) is a valid *cover-sequence* chosen during execution of the SAT-attack algorithm, (R_3, R_4) is not. The tuple notation is used to represent the order of the row sets. With respect to the SAT-attack algorithm, a row set that eliminates incorrect key columns not previously removed corresponds to a *distinguishing input pattern* (*DIP*), which is an input pattern that produces at least one different output response from the logic locked circuit when two separate key patterns are applied from the set of all possible remaining candidate keys. By querying the oracle, the adversary determines which of the dissimilar output patterns are incorrect and eliminates the corresponding incorrect keys.

For each iteration of the SAT-attack algorithm, the next DIP is determined through a subroutine call to a SAT solver. The algorithm then queries the oracle using the determined DIP and updates the set of remaining candidate key patterns based on the outputted response from the oracle. The process continues until only the correct key patterns remain. Therefore, the overall time-to-solve of a SAT-attack instance deobufscating a logic-locked circuit is approximately $\lambda \times t_{SAT}^{avg}$, where λ is the number of iterations and t_{SAT}^{avg} is the average execution time of each SAT-solver call. The t_{SAT}^{avg} term is SAT-solver specific and depends on multiple factors including the locked circuit topology, the representation of the Boolean formula, and the properties and configuration of the SAT solver [15]. In contrast, λ depends on the characteristics of the search space and the branching preference of the SAT solver. Consequently, the scaling factor λ, which is the number of iterations of the SAT solver or, equivalently, the number of DIPs applied to the SAT solver, is used as a proxy measure of the search time complexity of the SAT-attack.

C. Multiple search paths and DIP-sequence tree

Enumerating all possible DIP sequences yields a map of the paths through the SAT-attack search space, which are visualized as a tree diagram as shown in Fig. 2. The tree nodes are valid DIP candidates given the current path through the DIP sequence. The node values are the input patterns represented by row numbers. For example, the left most path is given as the DIP sequence (R_1, R_2, R_3).

D. Capturing search-space characteristics with U-truth tables

The branching path of a given DIP sequence depends on the structure of the multiset S_R, which compactly constitutes a truth-table corresponding to whether or not an output pattern of a locked circuit is uncorrupted for a given input pattern and key pattern. The **u**ncorrupted output truth-table U is shown in Fig. 1, where the scalar truth values 1 and 0 are denoted as ✓and ✗, respectively, to differentiate the U-truth table from the standard logical truth-table of a circuit.

For a locked circuit and an oracle, the resulting U-truth table defines the possible SAT-attack search paths, while abstracting information regarding the circuit topology and functionality. The structure or, equivalently, the pattern of the data included in the U-truth table provides a characterization of the search space, and therefore, is also described as a *U-search space*.

III. Expected Search Complexity

While the objective of the minimum set cover problem is to find the minimum number of DIPs $|S_C|_{min}$ needed to determine all incorrect key patterns, the analysis provides a worst-case evaluation of the security of a circuit and is, therefore, overly conservative. For a fair evaluation of the security of a circuit, a statistical-based model of SAT-resilience

is developed that includes an estimate of the expected number of DIPs $\mathbb{E}[|S_C^{(tuple)}|]$, the probability distribution function, and statistical measures including the variance.

A. Probabilistic model of the SAT-attack search path lengths

For a stochastic implementation of the SAT-attack, the next DIP is picked at random dependent on the branching preference of the SAT solver. The required number of DIPs to cover all incorrect keys is then a random variable given as D. The DIP-sequence tree represents the *probability space*. The probability mass function (PMF) of D, given by $\mathbb{P}(D = d)$, is the sum of the probabilities of the search paths with the same length. With the PMF defined, corresponding statistics such as the mean and the variance are derived.

B. Estimating branching probability

Without further knowledge of the specific stochastic implementation of the SAT-attack, the branching probabilities are assumed to be equally likely. Consequently, the resulting metrics are considered estimates of the number of required DIPs. The resulting probability for each DIP sequence path is provided in Fig. 2 for an equally likely branching probability. Specific implementations of the SAT-attack that include advanced heuristics or modifications to the SAT-attack algorithm such as with double-DIP [16] are modeled by setting the branching probabilities to a biased value including zero. The basic SAT-attack described in [9] is comprised of all possible search paths.

C. Metric based on probability-to-solve

The cumulative distribution function (CDF) of a random variable D, given by $\mathbb{P}(D \leq d)$, intrinsically represents the probability-to-solve within d number of iterations or queries to the oracle. The closed-form expression of the probability-to-solve is of significance when analyzing $\mathbb{P}(Break)$ in cryptographic paradigms that explore provable security.

IV. Closed-form metrics for out-of-cone logic locking techniques

SAT-resilient logic locking techniques described in [2], [3], [11], [12] include an integrated subcircuit to generate flip signals that are XOR-ed with a subset of the primary outputs of the circuit. The locking technique is, therefore, considered out-of-cone as the primary logic cone remains isolated from the locking subcircuit. As a result, the corresponding U-spaces are independent of the topology and the functionality of the original circuit. In addition, the U-spaces of the out-of-cone logic locking techniques are often well-organized patterns that are suitable for derivation of closed-form expressions of the estimated expected computational complexity as a function of the key size $|\vec{K}|$. A statistical analysis of the SAT-resilience of tenacious and traceless logic locking (TTLock) [11] is described in this section. The analysis is performed on the small circuit shown in Fig. 3, in which the DIP sequence tree is the same as that of the locked and-tree topology shown in Fig. 2. The analysis is generalized for TTLocked circuits of any size. Consequently, the analytical method does not suffer from combinatorial explosion when enumerating all possible DIP sequences. The various closed-form expressions associated with a TTLocked circuit are listed in Table I.

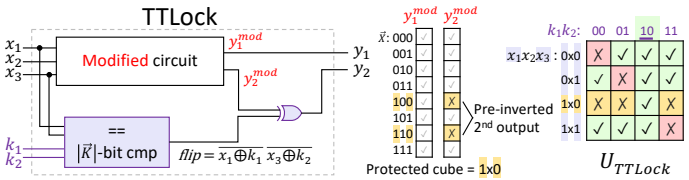

Fig. 3: An arbitrary 3-input by 2-output circuit locked by TTLock with an applied key size of 2. The corresponding U-space is also depicted.

TABLE I: Estimated PMF and statistics of the number of iterations D.

	TTLocked circuit, $\lvert\vec{K}\rvert \leq \lvert\vec{X}\rvert, N = 2^{\lvert\vec{K}\rvert}$
$\mathbb{P}(D = d) =$	$\begin{cases} 1/N & d \in \{1, 2, ..., N-2\}, \\ 2/N & d = N-1, \\ 0 & \text{otherwise} \end{cases}$ $\approx unif\{1, N\}$ as $\lvert\vec{K}\rvert \to large$
D_{min}	1
D_{max}	$N-1$
$\mathbb{E}[D]$	$\frac{(N-1)(N+2)}{2N} \approx \frac{N}{2} = 2^{\lvert\vec{K}\rvert - 1}$, as $\lvert\vec{K}\rvert \to large$
$\mathrm{Var}(D)$	$\frac{N^2-13}{12} + \frac{2}{N} - \frac{1}{N^2} \approx \frac{2^{2\lvert\vec{K}\rvert}}{12}$, as $\lvert\vec{K}\rvert \to large$

The resulting DIP-sequences of the TTLock U-space are equivalent to a *sampling without replacement problem*, where there are $N = 2^{|\vec{K}|}$ balls of which there is one red ball R and $N-1$ white balls W. The sampling game continues until either the red ball, which represents a protected cube, is drawn or all the white balls, which represent un-protected cubes, are drawn. Each ball represents a distinguishing cube pattern, such as 0x0 for example. Assuming an equally likely branching probability, the likelihood that the SAT-attack selects, for example, 3 DIPs to solve for a correct key is $\mathbb{P}(\text{draw } W, W, R) = \frac{N-1}{N}\frac{N-2}{N-1}\frac{1}{N-2} = \frac{1}{N}$. Whereas the probability of requiring the maximum number of DIPs of $N-1$ is $\mathbb{P}(\text{draw } N-2\ W\text{'s, then draw } R \text{ or the last } W) = \frac{N-1}{N}...\frac{2}{3}(\frac{1}{2} + \frac{1}{2}) = \frac{2}{N}$. The PMF is similar to the discrete uniform distribution $unif\{1, N\}$ with the difference being the merging of the $N-1$ and N cases.

V. Security gain per inserted key gate for in-cone logic locking

For most in-cone logic locking techniques predating the emergence of the SAT-attack [1], [17]–[19], key gates are inserted into the original circuit netlist or replace existing gates based on a given key gate placement strategy, which can be random [1] or guided by heuristics [18]–[20]. One objective when implementing an obfuscation technique is to minimize the number of inserted key gates while achieving the greatest SAT-resilience, which motivates a characterization of the trade-offs between the marginal overhead in area, power, and performance and the marginal security gain per inserted key gate. The gain in marginal resilience is described in this paper. A second objective is the characterization of the reduction in the marginal gain in the resilience of a circuit to the SAT-attack as the number of key gates increases, which indicates an optimal number of inserted key gates that results in the maximum achievable security against the SAT-attack. A third objective is to determine the optimal placement of key gates, given a target key size, that maximizes the SAT resilience of the circuit.

As a case study, a full-adder is locked by the random logic locking (RLL) technique [1], as shown in Fig. 4. An XOR/XNOR key gate is inserted either at an input of the gate or at a primary output. The proposed model and equally likely expected SAT-resilience estimator are used to quantify the effect that a selected location for placement of a key gate has

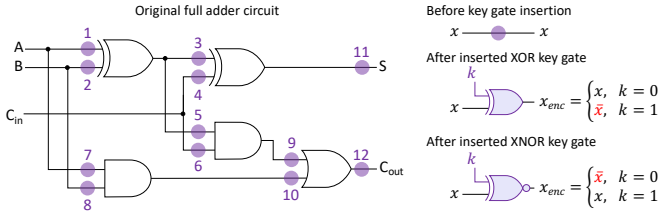

Fig. 4: Original full adder circuit with possible locations for the insertion of XOR/XNOR key gates shown with circular dots and corresponding node IDs.

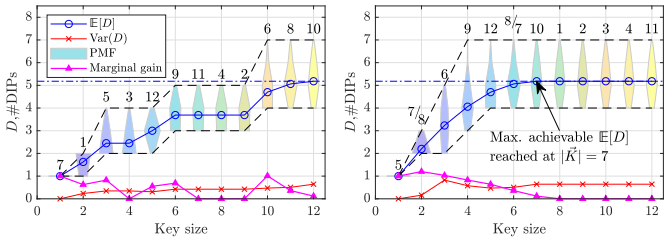

Fig. 5: Estimated expected SAT-attack resilience as a function of the key size and the placement of key gates accounting for both location and order for (a) a single randomly picked insertion sequence and (b) an insertion sequence that maximizes marginal gain.

on the SAT-resilience of a circuit, which provides insight on the nature of the security of in-cone logic-locking techniques.

Unlike out-of-cone logic locking techniques, the U-space of an in-cone logic-locked circuit depends on the topological structure of the original circuit. Consequently, the trend in the resilience of one circuit with respect to the key size does not necessarily apply to other circuits. In addition, for the given U-space, derivation of a closed-form expression of the expected SAT-resilience is a challenge. As a result, an approach that automatically enumerates all DIP sequences is required. Note that the number of DIP sequence combinations grows exponentially as the number of primary inputs is increased. Therefore, the use of a smaller circuit minimizes computational cost while allowing for the characterization of the effect that key gate insertion has on the marginal resilience.

To emulate an equally likely DIP selecting behavior, the entire truth-table of the U-space is first generated and each valid DIP is selected through a depth-first-search approach. For each DIP, the columns of incorrect key patterns are removed from the U-space truth-table. The algorithm is implemented in MATLAB and does not use any SAT-solvers. The current approach is simple and memory inefficient, but provides an initial characterization of the SAT resilience of a circuit.

A. Resilience as a function of key gate insertion sequence

The number on top of the PMFs shown in Fig. 5 is the location ID of a newly inserted key gate for an increase in the key size by one. The paths through the key gate insertion space, therefore, represent the cumulative resilience of the locked circuit as key gates are inserted. Two different SAT-resilience curves are shown in Fig. 5, which are a result of selecting dissimilar insertion sequences.

B. Resilience as non-decreasing function of key size

SAT-resilience is observed to be a monotonically increasing function with respect to the number of inserted key gates. Although, in the worst case, the resilience of the circuit to the SAT-attack remains unchanged for a sub-optimal placement of key gates, there is an undesired overhead in area, power, and performance due to the additional gates.

C. Maximum achievable expected resilience

For a given logic locking technique, gate insertion strategy, and original circuit topology, there exists a saturation in the security of the circuit beyond which no gains are observed. The maximum achievable expected resilience $\mathbb{E}[D]$ for both a random selection of insertion nodes and node selection based on maximum marginal gains is shown in Fig. 5a and 5b, respectively. Inserting all key gates results in the maximum achievable $\mathbb{E}[D]$; however, some insertion sequences yield the same level of security with a fewer number of inserted gates. As shown in Fig. 5b, the maximum resilience is already achieved for a key length of 7. Note that D_{max} is $7 < 8 = 2^{|\vec{X}|}$ even when all locations are utilized.

D. Most efficient key gate placement as a function of key size

If for each insertion decision the selected location providing the greatest resilience is chosen iteratively, then the most efficient sequence of inserted key gates is achieved as shown in Fig. 5b. For the full adder, the two primary insertion sequences that are the most efficient are (5,**8**,6,9,12,**7**,10,1,2,3,4,11), and (5,**7**,6,9,12,**8**,10,1,2,3,4,11). The gain in resilience diminishes as gates are added along the optimal path, which implies the approach of the saturation of key gates. The plateau due to the insertion of key gates at nodes 1, 2, 3, 4, and 11 reveals zero marginal gain, as these nodes are along the path that includes the two XOR gates of the full adder. Locking the input of an XOR gate using XOR-type key gates tends to yield duplicate columns in the U-space.

VI. Conclusions

The proposed probabilistic model of the SAT-attack search space and the equally likely statistical estimator are used to evaluate and quantify the SAT-resilience of a circuit obfuscated by both out-of-cone and in-cone logic locking techniques. The analysis is unbiased to the specific implementation of the SAT-attack, unlike methods based on experimental evaluation that include intrinsic bias. For out-of-cone locking, specifically TTLock, closed-form expressions as a function of key size for the minimum, maximum, mean, and variance of the number of DIPs are derived. For in-cone locking methods, where closed-form expressions are difficult to derive, simulation on a small circuit is performed, providing insight for key gate placement strategies. The first observation is that SAT resilience is a monotonically increasing function of the number of inserted key gates. Second, there exists a maximum achievable expected resilience constrained by the topology and the locking scheme. Finally, there exists a greedy algorithm that guides key gate insertion that results in the most efficient placement of key gates.

Acknowledgments

This research is supported in part by the Drexel Ventures Innovation Fund, the Air Force under Contract LXS016218, and 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] 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.

[3] 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.

[4] K. Juretus and I. Savidis, "Increasing the SAT Attack Resiliency of In-Cone Logic Locking," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2019.

[5] 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.

[6] 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.

[7] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, Aug. 2017. Advance online publication. doi: 10.1109/TETC.2017.2740364.

[8] M. El Massad, S. Garg, and M. V. 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] A. Sengupta, M. Nabeel, M. Yasin, and O. Sinanoglu, "ATPG-Based Cost-Effective, Secure Logic Locking," *Proceedings of the IEEE VLSI Test Symposium (VTS)*, pp. 1–6, Apr. 2018.

[14] 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.

[15] A. Biere and M. J. Heule, "The Effect of Scrambling CNFs," *Proceedings of Pragmatics of SAT*, Vol. 59, pp. 111–126, Jul. 2019.

[16] 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.

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

[18] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design & Test of Computers*, Vol. 27, No. 1, pp. 66–75, Jan. 2010.

[19] S. Dupuis, P. Ba, G. Di Natale, M. Flottes, and B. Rouzeyre, "A Novel Hardware Logic Encryption Technique for Thwarting Illegal Overproduction and Hardware Trojans," *Proceedings of the IEEE International On-Line Testing Symposium (IOLTS)*, pp. 49–54, Jul. 2014.

[20] K. Juretus and I. Savidis, "Characterization of In-Cone Logic Locking Resiliency Against the SAT Attack," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Jun. 2019. Advance online publication. doi: 10.1109/TCAD.2019.2925387.