Quality Assessment of Logic Locking Mechanisms using Pseudo-Boolean Optimization Techniques

Marcel Merten* Muhammad Hassan*† Rolf Drechsler*†

*University of Bremen, Germany
{mar_mer,hassan,drechsle}
@informatik.uni-bremen.de

†Cyber-Physical Systems, DFKI GmbH
28359 Bremen, Germany

***Abstract*—Nowadays, the manufacturing of *Integrated Circuits* (ICs) is highly distributed over different foundries yielding untrustworthy supply chains. This circumstance leads to concerns regarding the security, privacy, and reliability of the fabricated ICs, e.g., malicious usage and counterfeiting. *Logic Locking* (LL) is a prominent protection technique to safeguard against such concerns. Recently, the emerging technology of Reconfigurable *Field-Effect Transistors* (RFETs) has been utilized to implement new mechanisms based on *Polymorphic Logic Gates* (PLGs) to protect *Intellectual Property* (IP). The mechanisms' assessment is indispensable to reinforce the newly introduced logic obfuscation and, hence, avoid any security breaches. So far, formal SAT-based and approximate *Hamming Distance* (HD)-based assessment techniques have been used for determining the protection quality. While the approximate and formal approaches can detect many security threats [1], they are still unable to detect optimization-based attacks. This work proposes a novel formal approach based on *Pseudo Boolean Optimization* (PBO) to assess the quality of LL structures for sequential circuits, enabling the detection of currently unconsidered security breaches. In particular, the proposed approach leverages formal techniques to analyze the key and state space of a sequential circuit to evaluate the security against optimization-based attacks. The experimental evaluation validates that the proposed scheme unveils weaknesses of the protection structure, which remain undetected when using existing techniques.**

I. Introduction

Due to the distribution of *Integrated Circuits* (ICs) manufacturing, designers can benefit from access to advanced technology nodes without having the large capital expenditure of operating their semiconductor foundries. The distribution of the chips' manufacturing is one of the main security challenges due to a growing threat of compromising the integrity of once-trusted IC processes [2]. During the last decade, *Complementary Metal-Oxide-Semiconductor* (CMOS)-based protection mechanisms have been the dominant technology for implementing various protection measures. However, the layout-level obfuscation using CMOS-based camouflaging can introduce a large overhead concerning the required area and the resulting power consumption [3].

This work was financially supported by the German Federal Ministry of Education and Research BMBF under the framework of VE-CirroStrato. We would like to thank Verific Design Automation Inc. for providing the SystemVerilog frontend used for the implementation of our technique.

979-8-3503-3277-3/23/$31.00 ©2023 IEEE

Recent works like [2], [4], [5] have been focusing on achieving high protection while still preserving a low overhead by utilizing *Reconfigurable silicon nanowire Field-Effect Transistor* (RFET)-based *Polymorphic Logic Gates* (PLGs) [2], [6]–[8]. In [2], the quality of the resulting *Logic Locking* (LL) functionality is assessed by a simulation-based approach using a metric based on the *Hamming Distance* (HD) of the outputs. Therefore, the result is considered optimal if the HD is 50% of the maximal HD. On the other hand, formal approaches have been proposed, such as [1], showing the limitations of simulation-based approaches by unveiling further weaknesses in the protection mechanisms. In [1], the keys are only assessed using input patterns that result in a functional equivalence on all the *Primary Outputs* (POs). However, functional equivalence to the correct behavior is not mandatory in modern attack scenarios, such as approximate attacks. Contrary to other approaches approximate attacks, e.g. optimization-based attacks, exploit the search space to yield a key that results in behavior correct enough to consider the circuit as unlocked. Therefore, detailed information about the resulting encryption regarding individual POs is required to accurately unveil low corruption of the functional behavior and analyze the key space to evaluate the protection against optimization-based attacks.

This work proposes a novel approach to assess the quality of PLG-based and CMOS-based LL protection mechanisms by heavily utilizing the *Boolean Satisfiability* (SAT) problem and *Pseudo Boolean Optimization* (PBO) techniques to address the shortcomings of existing approaches. For the first time, the protection against optimization techniques is assessed by analyzing the exhibition of the correct functional behavior in the key space. The proposed technique combines the benefits of the existing HD-based and formal approaches, considering both the functional equivalence on single POs and the full state space. As a result, the proposed approach determines more information about the encrypted circuit's behavior and outperforms any existing analysis capabilities for weak LL mechanisms. In contrast to other techniques, all possible input pattern combinations are considered for the evaluation of the maximal functional equivalence given an incorrect key. Various experiments have been conducted on the ITC'99 benchmark set. The results prove that the proposed framework allows assessing PLG-based LL mechanisms effectively.

The remainder of this work is structured as follows: Section II briefly introduces the preliminaries as required for the comprehension of this work. Section III describes the proposed assessment scheme in detail, and Section IV presents the experimental evaluation. Finally, a conclusion and an outlook for future work are given in Section V.

II. Preliminaries

Within the last decade, much research work has been conducted to enhance electronic systems further while the classical CMOS technology has exceeded its physical limits. Reconfigurable technologies have gained a lot of interest in realizing more complex systems by employing PLGs [6]–[8].

A. Logic Locking

A well-known approach to prevent reverse engineering, even given the entire layout, is about introducing LL mechanisms. LL introduces a secret key to obfuscate the correct functional behavior of the circuit. However, several approaches exist for unlocking protected circuits. For instance, an SAT-based attack is a frequently used approach that utilizes SAT-solving techniques to ensure equivalent behavior to an unlocked circuit. Previous research like SARLock [9], Anti-SAT [10], TTLock [11], SFLL-HD [11] and SFLL-Rem [12] has focused on achieving severe resilience against SAT-based attacks. All beforementioned techniques achieve a high SAT resilience by introducing corruption only on a few inputs given an arbitrary incorrect key. As a result, the encryption of SARLock or similar approaches yields no proper IP protection [13]. For example, in [13] the encryption resulting from SARLock is considered unlocked since given each incorrect key the output is corrupted on only a single possible input. Therefore, approximate attack scenarios have been developed, e.g. optimization-based techniques, to detect keys that behave correctly enough to consider the circuit unlocked. Furthermore, all beforementioned LL techniques are based on sophisticated logical structures like perturbation- and restoration units, which are vulnerable to structural attacks. These attacks analyze the circuit structure to detect the LL mechanism, for example, to remove it. Recent combinations of structural and functional attacks like SPI [14] or Valkyrie [15] have been able to take advantage of that weakness to break these LL techniques. Such resistance against structural attacks can be achieved if the LL mechanism is indispensable to obtaining the correct functional behavior.

To introduce LL gates without the high-performance overhead of CMOS-based techniques, PLGs can replace gates of the original circuit, which has a high impact on the primary outputs [2]. Different concepts have been proposed to realize a device-level reconfiguration with PLGs. PLGs are an effective way to implement a LL mechanism since they realize multiple functionalities in the same cell, whereby the actual functionality is chosen by configuring a control signal.

B. Quality Assessment of RFET-based Logic Locking Mechanisms using Formal Methods

This paragraph briefly introduces the SAT-based quality assessment approach of [1]. The *Circuit-under-Assessment* (CuA) is analyzed for corrupting keys - incorrect keys that result in correct functional behavior given at least one input pattern. Therefore, a miter circuit is created from the *CuA* considering the correct key k_c - yielding the *Conjunctive Normal Form* (CNF) Φ_{k_c} - and any incorrect key $\widehat{\mathcal{K}}$ resulting in $\Phi_{\widehat{\mathcal{K}}}$. For considering sequential elements, the CuA is unrolled for *N* clock cycles. The *Flip-Flops* (FFs) are modeled as *Pseudo Primary Inputs* (PPIs), initialized with 0.

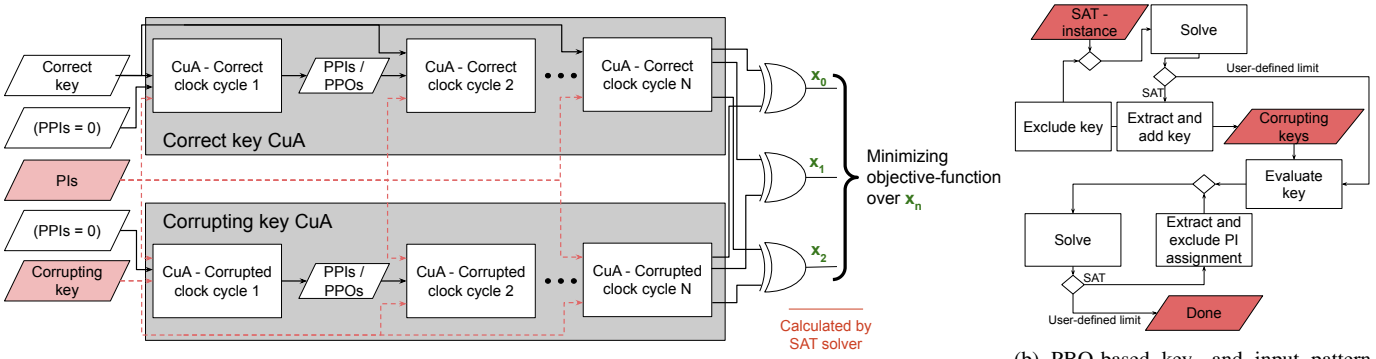
The entire model is stored as one CNF Φ_{comp} and processed by a state-of-the-art SAT solver. Consequently, the introduced inverted miter compares the unrolled Φ_{k_c} with the unrolled $\Phi_{\widehat{\mathcal{K}}}$, i.e., considering any incorrect key $k_i \neq k_c, k_i \in \widehat{\mathcal{K}}$. More precisely, both the state – defined by the stored FFs' values – and the primary output values can be compared for all N observed clock cycles. If a satisfiable solution is determined, a corrupting key k_f has been detected, yielding a functional equivalent of the CuA given at least one input pattern.

For a qualitative assessment of the discovered security threat, every determined corrupting key k_f is evaluated against possible input patterns leading to functional equivalence regarding this breach. More precisely, the individual corrupting key is enforced in $\Phi_{\widehat{\mathcal{K}}}$, by additional clauses. The key detection procedure – including the security threat evaluation regarding the discovered corrupting key – is repeated until Φ_{comp} is unsatisfiable, meaning that there are no more corrupting keys to be detected or a user-defined limit has been exceeded.

C. Optimization-based Attacks on Logic Locking Systems

As an approximate attack scenario, optimization-based attacks define the determination of a key as an Optimization problem. An optimization problem is typically defined as a riple $(\Omega, \lambda, \prec)$, with Ω the search-space, $\lambda : \Omega \rightarrow \mathbb{R}$ the fitness function and $\prec \in \{>, <, =\}$ a relational operator. Given a logic locked circuit with an unknown k_c, an optimization problem can be defined to maximize the functional equivalence to the unlocked circuit. The search space is given overall $k \in K$. Many approaches exist for examining the search space of an optimization problem, like *Evolutionary Algorithms* (EA). The general idea of EAs is simulating the natural evolution process to solve an optimization problem [1].

A hardware-based EA attacking the correct key of a locked system called GenUnlock has been proposed in [18]. All of the in [18] considered LL mechanisms were unlocked in less than 1,000 seconds using a hardware-based GenUnlock approach, yielding a serious security threat. One of the most important parts of an *EA* is the fitness function which evaluates a solution – in this specific problem, a key – within the search space. In [18], the fitness is reflected by the equivalence to the correct functional behavior. The search space contains globally optimal solutions (global optima), which evaluate the maximal fitness within the search space.



(a) Resulting abstract model of the CuA

(b) PBO-based key- and input pattern extraction

Figure 1: Quality assessment technique

If the correlation between the fitness of a solution and its distance to the global optima is a (perfect) negative correlation, the fitness structure of the search space is (monotonously) guiding. Given a monotonously guiding fitness structure, an approximation of a global optimum is trivial.

III. Quality Assessment Framework

This section describes the generation of the PBO-based model for the quality assessment of a CuA using PLG-based LL mechanisms.

Following the basic model of [1], a comparing miter structure is generated from the CuA while considering the a-priori known *correct* key k_c yielding the CNF Φ_{k_c} and any incorrect key in $\widehat{\mathcal{K}}$ yielding $\Phi_{\widehat{\mathcal{K}}}$. The basic principle of this construction is given in Sub-figure 1a. The CuA is unrolled for N clock cycles since sequential elements – meaning FFs – have to be considered for an exact assessment in terms of sequential circuits' unrolling [19]. Here, the value N has to be adjusted for the CuA characteristics. Depending on the CuA, the number of clock cycles required to achieve the relaxation of the circuit given an input pattern varies. Similar to the approach proposed in [1], 0 is assumed as the initialization value for all FFs in cycle $i = 1$.

The FFs are modeled as *Pseudo Primary Inputs* (PPIs) in cycle $i + 1$ and are connected to the corresponding *Pseudo Primary Outputs* (PPOs) of the previous cycle i. Furthermore, the primary inputs are equally driven for both unrolled instances (of the CuA) and are kept constant during the unrolling. After the miter structure has been added, the key is constrained for both instances of the unrolled CuA. For Φ_{k_c} , the correct key k_c is set by adding clauses implying k_c, whereby $\Phi_{\widehat{\mathcal{K}}}$ is extended by a conflict clause excluding k_c. The entire model is stored as one CNF Φ_{comp} and processed by a state-of-the-art PBO solver. The introduced miter structure compares the unrolled Φ_{k_c} with the unrolled $\Phi_{\widehat{\mathcal{K}}}$, i.e., considering any incorrect key $k_i \neq k_c, k_i \in \widehat{\mathcal{K}}$. More precisely, both the state and the primary output values can be compared for all N observed clock cycles. Compared to the formal approach of [1], the proposed approach is more precise in the assessment of an incorrect key. While in [1], the assessment tool is only able to distinguish between functional (in) equivalences, the proposed assessment framework can maximize the functional equivalence on the POs given an incorrect key.

In particular, an objective-function θ defined in Equation 1 is maximizing the logical equivalence is defined over the XOR-gates $x_i \in X$ comparing Φ_{k_c} and $\Phi_{\widehat{\mathcal{K}}}$.

$$\theta = min(\sum_{x_i \in X} (x_i * 1)) \tag{1}$$

Sub-figure 1b presents the general approach for evaluating incorrect keys. Φ_{comp} is iteratively solved with the optimization target to minimize the inequivalence on the POs. Compared to the assessment tool of [1], Φ_{comp} does not become unsatisfiable, that is due to the optimization approach. The PBO-solver iteratively collects key and input pattern pairs that had the maximum equivalent functional behavior and removes the collected incorrect key k_f from the search space. Each satisfying solution provides an incorrect key k_f yielding the maximum achievable functional equivalent behavior given at least one input pattern. This circumstance indicates that k_f has a high potential to form a major security breach. For later analysis of the protection against optimization-based attacks, the *HD between the Correct key k_c and the detected incorrect key k_f* (HDCI) is calculated. The algorithm also determines whether further incorrect keys exist that exhibit the circuit's correct functional behavior. Thus, the evaluated k_f is excluded by adding a conflict clause to $\Phi_{\widehat{\mathcal{K}}}$. The key detection procedure is repeated until a predefined number of keys is examined.

For a qualitative assessment of the discovered security threat, every determined incorrect key k_f is evaluated against a predefined number of possible input patterns leading to a maximum functional equivalence regarding this breach. More precisely, the individual incorrect key is enforced in $\Phi_{\widehat{\mathcal{K}}}$, by additional clauses. Afterward, Φ_{comp} is re-evaluated to detect the input pattern resulting in a maximum equivalent behavior using k_f and, hence, the solving process is repeated iteratively. After each iteration, the most problematic input pattern calculated by the SAT solver is extracted for later analysis and excluded from the further search process of the PBO solver. The process ends as soon as a user-defined limit has been exceeded. As a result, input patterns with totally correct behavior are detected similar to the approach proposed in [1]. However, the PBO-based approach allows the detection of additional input patterns that have the lowest functional

equivalence on the POs. Therefore, analytical metrics like the *Average functional Equivalence on the POs* (AEP) of the detected keys and input patterns can be applied to unveil low corruption of the functional behavior or POs that are rarely affected by the applied LL. The enhanced assessment capabilities allow the analysis of the key space for each captured incorrect key k_f, by calculating the *Correlation between HDCI and AEP* (CHA).As a result, the protection against optimization-based attacks can be analyzed to evaluate the threat of keys that unlock the majority of the circuit's behavior. In conclusion, the proposed approach allows determining incorrect keys and evaluates their threat to the protection system while not relying on total functional equivalence given a key and input pattern pair. Therefore, detailed information about the encryption of the protection mechanism is obtained, yielding superior analysis capabilities compared to previous techniques.

IV. Experimental evaluation

This section describes the experimental evaluation of the proposed PBO-based quality assessment framework for PLG-based LL protection mechanisms and discusses the obtained results. The conducted experiments are clearly distinguished against existing formal (SAT-based) [1] and simulation-based (HD) [2] approaches used for comparison as the baseline.

All experiments have been executed on an *AMD 4750U* processor with 40 GB system memory. The proposed technique has been solely implemented in C++ using yices2 for PBO. Different benchmark circuits of the *ITC'99* benchmark suite are considered for the evaluation. Next, a relevant LL approach needs to be inserted. A meaningful LL mechanism has to be able to address the resulting encryption and the resilience against state-of-the-art attacks. The main goal of LL is the encryption of the functional behavior if an incorrect key is applied. However, increasing the resilience against SAT-based attacks usually minimizes the encryption of the LL mechanism. In conclusion, the inserted protection mechanism should provide a trade-off between SAT resilience and the resulting encryption. Furthermore, the resistance against structural attacks is of utmost importance to prevent removal attacks. Therefore, the LL mechanism is applied by replacing CMOS gates in the original circuits with PLGs that obtain the same functionality if correctly configured. The specific placement of the PLG-based LL influences the encryption and SAT resilience. Additionally, the PLG-based LL is an inherent part of the circuit, with no possibility to remove it without knowing the correct key. Utilizing PLGs is essential to significantly reduce the resulting overhead and, hence, increase the relevance of this approach to LL. Therefore, the proposed quality assessment framework is evaluated on circuits with a PLG- and replacement-based LL mechanism. For each of the considered circuits, 100 of the *NOR*, *NAND*, *XOR*, and *XNOR* gates have been randomly replaced by PLGs, while the functional behavior is retained if the correct key is applied. Consequently, each circuit has 100 control signals resulting in 2^{100} possible keys. The random placement of the PLGs does not ensure optimal encryption or resilience against attack scenarios. However, for evaluating assessment frameworks, experimental evaluations have shown that 100 PLGs can be considered a sufficient number of key gates to create LL structures with weaknesses that are difficult to analyze and, hence, hard to detect. Furthermore, a large key space is obtained, yielding a non-trivial attack scenario. Furthermore, the 1,024 input patterns with the most functionally correct behaving POs (per incorrect key k_f) are captured.

For the consideration of a simulation-based approach, the HD-based approach proposed in [2] has been enhanced to enable the assessment of sequential circuits. The CuA is simulated for five clock cycles, whereby the FFs are initialized with 0, and the input pattern is kept constant over all five clock cycles. Each circuit has been unrolled for five clock cycles using the SAT-based- as well as the proposed PBO-based approach. This number of cycles has been proven as an appropriate parameter to cover the functional behavior's majority (of the considered benchmark circuits) [20].

Table I shows the detailed results of the HD-based baseline approach, the SAT-based baseline approach proposed in [1], and the novel PBO-based approach. The average $HD_{avg}(S)$ over ten considered simulated sets S is calculated for the HD-based approach. Here, each simulation run contains 10,000 randomly chosen input patterns and key combinations, i.e., $(pi, k) \in S$ with $|S| = 10,000$. The formal baseline approach considers the absolute number of identified corrupting keys, the minimum, the average, and the maximum number of corrupted input patterns per key. The novel approach considers the same evaluation criteria, with the difference that a user-defined number $\#K = 1,023$ incorrect keys is assessed. More precisely, the $\#K$ keys resulting in the highest equivalence to k_c given at least one input pattern are assessed. Since all considered circuits have at least 1,023 corrupting keys, the results are equivalent for the considered benchmarks and condensed for both approaches in Table I. In addition, the AEP out of the conducted data is calculated, reflecting the functional equivalence between the correct and the considered incorrect keys. If a *Monotonic Decreasing HDCI by a Monotonic Increasing AEP* (MDI) is given, no local optima are given in the fitness function of a potential optimization-based attack. Even if such MDI is not given, the CHA can provide valuable information about the guiding structure of the search space.

Regarding the execution time, the proposed approach is 6.63× higher than the SAT-based approach. The PBO-based optimization process results in comparatively high, but still reasonable, execution times with increasing circuit complexity. Furthermore, the assessment framework is evaluated with a key space of 2^{50} keys to assessing the scalability regarding the number of possible keys. Due to the page limitation, these evaluations have not been visualized. However, the evaluation shows an average time overhead of only 3% with increasing the number of control signals from 50 to 100. The time overhead of the proposed approach mainly results from the calculation of the most equivalent behaving input patterns given an incorrect key. Since the key is given, execution time is not affected by the key size. Therefore, the proposed approach proves great scalability with the number of possible keys. Next, the assessment capabilities of the considered approaches are compared.

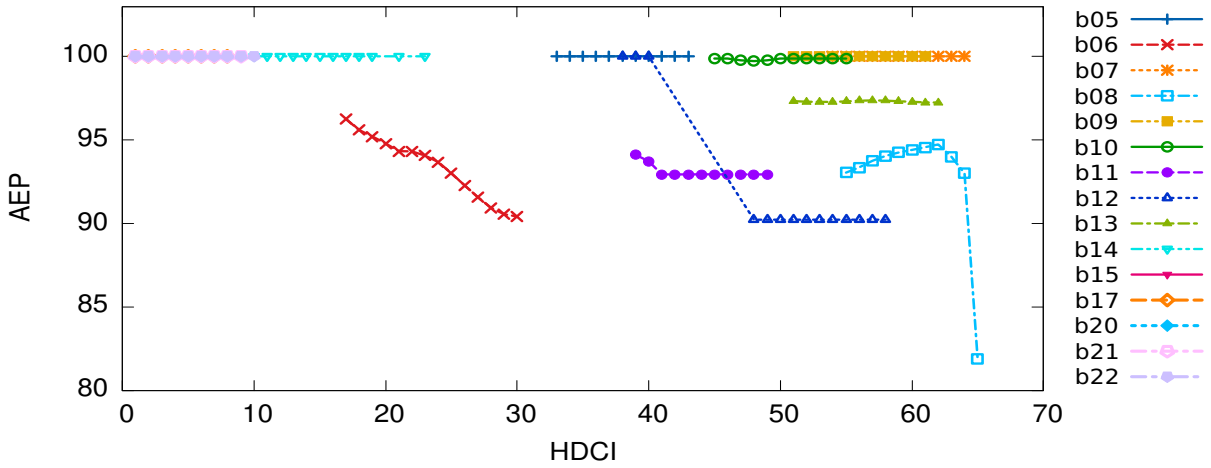


Figure 2: Correlation between HDCI and AEP (CHA) for detected incorrect keys

TABLE I: Comparison results of HD-based assessment [2], SAT-based approach [1] and newly proposed PBO-based method

Circuit	**HD [2]**	**SAT-based [1] and PBO-based approach**				**PBO-based approach**			**time [s]**		
		$\#\{k_f\}$	#correct behaving input pattern			AEP	MDI	CHA	HD [2]	SAT-based [1]	PBO-based
			minimum	average	maximum						
b05	0.07	1,023	2	2	2	100.00	-	-	223	28	28
b06	0.57	1,023	1	1.01	2	90.04	1	-0.99	28	4	2
b07	0.13	1,023	2	2	2	100.00	-	-	182	20	20
b08	0.55	1,023	1	254.98	256	92.27	0	-0.45	84	159	159
b09	0.53	1,023	2	2	2	100.00	-	-	62	7	7
b10	0.22	1,023	64	953.73	960	96.23	0	0.38	109	21	717
b11	0.10	1,023	2	2.01	4	93.53	1	-0.67	190	165	164
b12	0.12	1,023	16	16.16	32	95.11	1	-0.90	382	144	145
b13	0.46	1,023	512	512	512	97.15	0	-0.28	138	969	986
b14	0.01	1,023	1,024	1,024	1,024	100.00	-	-	1457	4162	15181
b15	0.01	1,023	1,024	1,024	1,024	100.00	-	-	2228	2278	23722
b17	0.01	1,023	1,024	1,024	1,024	100.00	-	-	6409	3227	68715
b20	0.01	1,023	1,024	1,024	1,024	100.00	-	-	2971	2886	28218
b21	0.06	1,023	1,024	1,024	1,024	100.00	-	-	2907	2974	27592
b22	0.01	1,023	1,024	1,024	1,024	100.00	-	-	4452	7150	44758

Following the HD-based approach, the circuits b06, b08, b09, and b13 are close to the $HD_{avg}(S) = 0.5$, considered optimal in [2]. The existing SAT-based quality assessment technique and the novel PBO-based approach show that all these circuits have at least one k_f resulting in equivalent circuit behavior. In the case of b09, all corrupting keys lead to an equivalent circuit behavior compared to the one when applying the correct key. In particular, b09 has at least 1,023 corrupting keys, unlocking the circuit, while the $HD_{avg}(S)$ of 0.53 suggests alleged good protection. This reflects once more the very low quality of the underlying LL mechanism and the complete misjudgment of the HD-based assessment.

However, the SAT-based approach [1] detects only total equivalent functional behavior given a key and input pattern pair. This circumstance can result in a misleading assessment of the LL mechanism quality as well. For example, for the b13 candidate, all detected keys yield a total correct functional behavior on half of the assessed input patterns. However, the SAT-based approach does not indicate the equivalent behavior for the other possible input pattern. The novel PBO-based technique reveals a major security breach while detecting an average correct functional behavior of approx. 97.15% POs, given the detected incorrect keys and the corresponding most equivalent behaving input pattern. Meaning the LL mechanisms only affects very few of the POs given an incorrect key. Additionally, this security threat remains undetected using the HD-based approach since the high equivalence on the POs can solely be observed on the rare most intimidating keys. However, for an arbitrary key, the $HD_{avg}(S)$ of 0.46 misleadingly indicates good encryption.

Beyond combining the benefits of the existing approaches, the novel PBO-based method also calculates the potential threat of optimization-based techniques, as achieved by considering the MDI. The MDI leads to an optimal fitness function for an optimization problem. The CHA can be visualized even if such MDI behavior is not necessarily given. Figure 2 shows the AEP and HDCI of the considered circuits, representing a visualized CHA. The correct key, the target of the optimization-based attack, can be imagined in the upper left corner at an AEP of 100% and an HDCI of 0. The circuits b05, b07, b09, b14, b15, b17, b20, b21, and b22 yield correct behavior regarding every assessed key and input pattern pair and, hence result

in an AEP of 100.00. Therefore, the metrics MDI and CHA are not calculated for these circuits since both are based on a correlation between the HDCI and AEP. Since b05, b07, and b09 only have two possible input patterns, there are at least 1,023 incorrect keys with total equivalent behavior to the correct key. In the case of b14, b15, b17, b20, b21, and b22, at least 1,024 input patterns are corrupted. The negative *CHA* of the b06, b11, and b12 provides a guiding structure for an exploration of the search space with optimization problem-solving techniques. Therefore the LL mechanisms provide weak protection against optimization-based attacks. Furthermore, in the case of the b06, b11, and b12, MDI is given, resulting in a trivial search space since there are no local optima. As a result, optimization-based attacks can easily determine a key, that unlocks the beforementioned circuits for all, or at least the majority, of possible inputs. This weakness against approximate optimization attacks remains undetected using the existing assessment approaches.

In conclusion, the proposed approach detects key and input pattern pairs resulting in low encryption of the circuit behavior to unveil serious security threats. For example, the proposed approach determines inputs resulting in 100% equivalence on the POs, while the HD of 0.53 misleadingly indicates a high corruption in the POs. Compared to the SAT-based approach [1], the proposed method leverages PBO-based techniques to detect the input pattern with the most correct behavior regarding the POs. Therefore, more detailed information about the affected POs is obtained that can be used for a detailed analysis, e.g. analyzing the threat of optimization problem-solving techniques. By this, existing quality assessment techniques are clearly outperformed, paving the way for highly effective protection mechanisms as required by nowadays applications.

V. Conclusions

This paper presented a novel method for assessing the quality of (PLG-based) LL protection systems by heavily orchestrating PBO techniques. In the end, the proposed framework allows determining incorrect keys and evaluates their threat to the protection system by calculating their functionally most correct behaving input pattern. The assessment combines the benefits of the SAT-based- and HD-based approaches and, hence allows to consider the full state space while not depending on the existence of corrupting keys. For the first time, the protection of PLG-based LL mechanisms against optimization-based attacks has been assessed. Future work will investigate a compositional approach allowing for processing even larger industrial-sized designs. Furthermore, an SAT-based reinforcement technique is being developed to use the obtained information and optimize the LL placement towards high encryption while preserving a high resilience against different attacks.

Acknowledgment

The authors gratefully thank Sebastian Huhn from Siemens Electronic Design GmbH (Germany) and Jens Trommer from NaMLab gGmbH for the inspiring research discussions and thoughtful feedback.

References

[1] M. Merten, S. Huhn, and R. Drechsler, "Quality Assessment of RFET-based Logic Locking Protection Mechanisms using Formal Methods," in *IEEE European Test Symposium*, 2022, pp. 1–2.

[2] Q. Alasad, J.-S. Yuan, and Y. Bi, "Logic locking using hybrid CMOS and emerging SiNW FETs," *Electronics*, vol. 6, no. 3, 2017.

[3] S. Rai, S. Srinivasa, P. Cadareanu, X. Yin, X. S. Hu, P.-E. Gaillardon, V. Narayanan, and A. Kumar, "Emerging reconfigurable nanotechnologies: Can they support future electronics?" in *IEEE/ACM International Conference on Computer-Aided Design*, 2018, pp. 1–8.

[4] Q. Alasad and J. Yuan, "Logic obfuscation against IC reverse engineering attacks using PLGs," in *IEEE International Conference on Computer Design*, 2017, pp. 341–344.

[5] Q. Alasad, J.-S. Yuan, and P. Subramanyan, "Strong logic obfuscation with low overhead against IC reverse engineering attacks," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 4, pp. 1–31, 2020.

[6] J. Trommer, A. Heinzig, S. Slesazeck, T. Mikolajick, and W. M. Weber, "Elementary Aspects for Circuit Implementation of Reconfigurable Nanowire Transistors," *IEEE Electron Device Letters*, vol. 35, no. 1, pp. 141–143, 2013.

[7] J. Trommer, A. Heinzig, T. Baldauf, S. Slesazeck, T. Mikolajick, and W. M. Weber, "Functionality-Enhanced Logic Gate Design Enabled by Symmetrical Reconfigurable Silicon Nanowire Transistors," *IEEE Transactions on Nanotechnology*, vol. 14, no. 4, pp. 689–698, 2015.

[8] T. Mikolajick, G. Galderisi, S. Rai, M. Simon, R. Behrle, M. Sistani, C. Cakirlar, N. Bhattacharjee, T. Mauersberger, A. Heinzig, A. Kumar, W. Weber, and J. Trommer, "Reconfigurable field effect transistors: A technology enablers perspective," *Solid-State Electronics*, vol. 194, p. 108381, 05 2022.

[9] 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*, 2016, pp. 236–241.

[10] Y. Xie and A. Srivastava, "Anti-SAT: Mitigating SAT Attack on Logic Locking," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 2, pp. 199–207, 2019.

[11] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, *Provably-Secure Logic Locking: From Theory To Practice*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017.

[12] A. Sengupta, M. Nabeel, N. Limaye, M. Ashraf, and O. Sinanoglu, "Truly Stripping Functionality for Logic Locking: A Fault-Based Perspective," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4439–4452, 2020.

[13] Y. Shen and H. Zhou, "Double DIP: Re-Evaluating Security of Logic Encryption Algorithms," ser. GLSVLSI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 179–184.

[14] Z. Han, M. Yasin, and J. J. Rajendran, "Does logic locking work with EDA tools?." *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1055–1072, 2021.

[15] N. Limaye, S. Patnaik, and O. Sinanoglu, "Valkyrie: Vulnerability Assessment Tool and Attack for Provably-Secure Logic Locking Techniques," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 744–759, 2022.

[16] S. A. Cook, "The complexity of theorem-proving procedures," in *ACM Symposium on Theory of Computing*. Association for Computing Machinery, 1971, p. 151–158.

[17] S. Huhn, S. Eggersglüß, K. Chakrabarty, and R. Drechsler, "Optimization of retargeting for IEEE 1149.1 TAP controllers with embedded compression," in *Design, Automation and Test in Europe*, 2017, pp. 578–583.

[18] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "GenUnlock: An Automated Genetic Algorithm Framework for Unlocking Logic Encryption," in *IEEE/ACM International Conference on Computer-Aided Design*, 2019, pp. 1–8.

[19] R. Arora and M. Hsiao, "Enhancing SAT-based bounded model checking using sequential logic implications," in *International Conference on VLSI Design*, 2004, pp. 784–787.

[20] A. Finder, A. Sülflow, and G. Fey, "Latency analysis for sequential circuits," in *IEEE European Test Symposium*, 2011, pp. 129–134.