# Increasing SAT-Resilience of Logic Locking Mechanisms using Formal Methods

Marcel Merten* Sebastian Huhn*† Rolf Drechsler*†

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

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

***Abstract*—Today, *Integrated Circuits* (ICs) manufactoring is distributed over various foundries, resulting in untrustworthy supply chains. Therefore, significant concerns about malicious intentions like intellectual property piracy of the fabricated ICs exist. *Logic Locking* (LL) is one well-known protection technique to improve the security of ICs. However, there are approaches to unlocking the circuit, like the SAT-based attack. Significant research has been done on thwarting the SAT-based attack by providing SAT-resilient LL. Nevertheless, these SAT-resilient LL approaches have an inherent structural footprint, yielding a high vulnerability to structural attacks. Recently, *Polymorphic Logic Gates* (PLGs) have been utilized to implement logic obfuscation by replacing gates. *Reconfigurable Field Effect Transistors* (RFETs) are a new emerging technology for implementing such PLGs due to their inherent camouflaging properties. This work proposes a novel technique for increasing SAT-resilience while introducing no structural weakness using those PLGs. In particular, based on the concept of an SAT-based attack, a procedure for determining the most SAT-resilient placement of LL-cells is developed. The experimental evaluation proves that the proposed hardening of the placement increases the SAT-resilience compared to a random placement while providing inherent camouflaging of RFET-cells.**

## I. Introduction

In the last decades, a lot of research has been conducted to address the major security challenges, like Intellectual Property (IP)-piracy. A famous approach for thwarting reverse engineering of the developed design is called *Logic Locking* (LL) or logic obfuscation. This approach introduces key gates to encrypt the functional behavior of an original circuit. Afterward, the design is protected by a secret correct key, yielding the correct functional behavior. As a result, LL prevents the potential security threat, even given the functional description of the circuit. Therefore, LL is a strong security measure protecting against powerful threats such as malicious misuse from untrusted manufacturers.

Early adoptions of logic obfuscation like [1], [2] provide a certain protection by implementing XOR or MUX-based key gates. These approaches yield high encryption of the circuit's functional behavior, while providing alleged good protection against brute force attacks by implementing a sufficiently large number of key gates.

However, with the development of the SAT-based attack [3], a novel threat of formal attacks arise. The SAT-based attack leverages the effectiveness of modern SAT-solving techniques to determine *Distinguishing Input Pattern* (DIPs). Combined with an oracle, for instance, an overproduced sample of the circuit using the correct key, the SAT-based attack potentially unlocks all aforementioned protection mechanisms.

The SAT-based attack [3] is one of the main concerns for LL encryption mechanisms. A lot of research has been conducted on thwarting the SAT-based attack, like SARLock [4], Anti-SAT [5], TTLock [6], S-URSAT [7], SFLL-HD [6] and SFLL-Rem [8]. However, all of the beforementioned LL mechanisms add sophisticated structures, e.g., perturbation and restoration units, making them vulnerable to structural attacks. Combined structural- and functional attacks, like the Fall Attack [9] or Valkyrie [10], leverage this weakness to unlock these LL techniques.

This work proposes a novel technique to introduce polymorphic logic gate-based LL protection mechanisms by heavily orchestrating the *Boolean Satisfiability* (SAT) problem to address the resilience against SAT-based attacks without adding a structural footprint. More precisely, the encryption logic is an inherent part of the functional behavior that can not be removed or bypassed. Therefore, a framework has been designed, allowing the insertion of a protection mechanism, by solely replacing the gates of an unlocked circuit with *Polymorphic Logic Gates* (PLGs). In particular, gates that provide a high SAT-resilience are determined by utilizing formal techniques.

Several experiments have been executed while considering the ITC'99[11] benchmark set. The results clearly prove the successful encryption by replacing existing *Complementary Metal-Oxide-Semiconductor* (CMOS)-based gates with PLGs using the proposed technique. Furthermore, the proposed placement determined by formal techniques outperforms the random placement of LL cells.

The remainder of this work is structured as follows: Section II briefly introduces previous research and clearly distinguishes them against this work. Section III describes the proposed LL placement scheme in detail, and Section IV

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

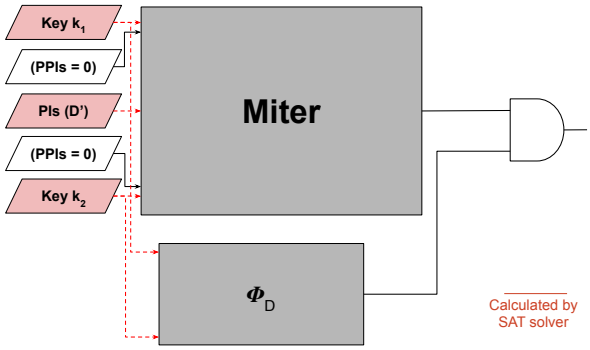



Fig. 1: Basic concept of a SAT-based attack

presents the experimental evaluation. Finally, a conclusion and an outlook on future work are given in Section V.

## II. Preliminaries

Within the last decade, a lot of research has been conducted to prevent the malicious misuse of IP components. With the SAT-based attack proposed in [3], it was easy to unlock early adoptions of logic obfuscation. As a result, many LL mechanisms have been developed to increase the resilience against SAT-based attacks. The encryption of the aforementioned protection mechanisms results from additional perturbation logic. Reconfigurable technologies have gained a lot of interest when realizing complex logic. This emerging technology proved promising to exceed the constraints of Moore's law by employing PLGs. For example, previous research like [7] used PLGs to significantly decrease the necessary overhead of the encryption.

### A. Polymorphic Logic Gates

Various concepts of PLGs for device-level reconfiguration has been proposed. A PLG [12], [13] can implement multiple functionalities within a single cell. Recent research like [14], [7] uses *Reconfigurable Field Effect Transistors* (RFETs) as PLGs to introduce efficient logic obfuscation. RFETs have a control signal that can be used for the configuration between n-channel and p-channel behavior [15], yielding the polymorphic behavior. New protection mechanisms like on-chip key storage can be implemented by leveraging the reconfiguration capabilities of this new emerging technology [16]. Furthermore, the RFET technology is also promising to implement effective protection mechanisms against side-channel attacks.

A well-known technique to prevent reverse engineering, even given the entire netlist, is LL-based encryption of an unlocked circuit. LL obfuscates the correct functional behavior with a secret key. CMOS-based approaches usually introduce MUX gates [17], [18], [19] of XOR/XNOR key gates [20], [21], [22] to encrypt the correct functional behavior of the original circuit. As a result, a huge overhead in the area- and power-consumption is the consequence [14].

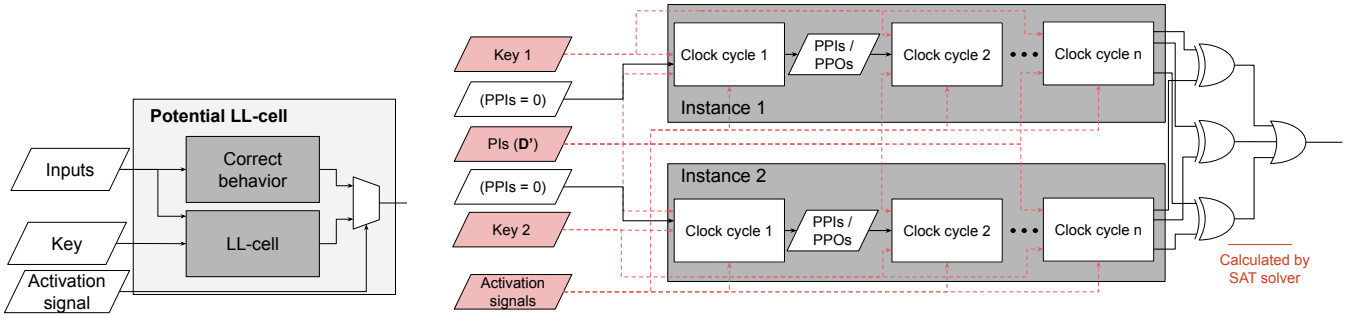
PLGs like RFETs are an effective way to implement a protection mechanism since they implement multiple functionalities in the same cell. Different RFET-based cells are available that implement multiple functionalities like NAND/NOR- or XOR/XNOR. The actual functionality of the gate is chosen by configuring a control signal [12], [13]. To insert key gates without the high performance overhead of CMOS-based techniques, PLGs can replace CMOS-based gates of the original circuit. In [14], this is done by replacing gates that have a high impact on the primary outputs. As a result, high encryption is obtained. However, the performance of threats like the SAT-based attack benefit from the high obfuscation.

### B. SAT-based Attacks

One of the first proven NP-complete problems is the *Boolean Satisfiability* (SAT) problem [23]. A lot of research on SAT-solving techniques significantly increased the effectiveness of solving SAT problems over the years. While camouflaging and logic obfuscation intends to prevent malicious intentions regarding intellectual property, attackers permanently try to develop techniques to unlock or remove such protection mechanisms. A frequently used attacking algorithm to unlock an encrypted circuit functionally is the SAT-attack, as proposed in [3]. The SAT-based attack leverages the effectiveness of SAT-solving techniques to unlock the circuit by determining the correct key $k_c$ or an equivalent behaving key. First, a miter structure of two instances of the encrypted circuit is instantiated. By solving the miter instance, a DIP $D$ and a pair of keys $(k_1, k_2)$ is calculated for the *Primary Inputs* (PIs). The DIP $D$ is an input pattern, which results in a differing output behavior using $k_1$ and $k_2$, hence at least one of the output behaviors of the two compared keys is incorrect. Next, an unlocked product $\Omega$ of the chip is used as an oracle, yielding the correct output behavior $\Omega(D)$ for the DIP $D$. Afterward, the key space of $k_1$ and $k_2$ is constrained to satisfy the correct output behavior $\Omega(D)$ for the previously calculated DIP $D$ by adding an SAT-instance $\Phi_D$ consisting of two inverted miters. Each inverted miter forces correct behavior using the logic locked circuit with the key $k_1$ or $k_2$, on the stimulus $D$. By solving the SAT instance again, the next DIP $D'$ is obtained and the procedure is repeated by adding $\Phi_{D'}$. The basic principle of the SAT-based attack is illustrated in Figure 1.

### C. SAT-Resilience and Structural Attacks

Due to the growing number of attack scenarios and techniques, LL is constantly adapted to thwart these attacks. One of the most intimidating attacks is the SAT-based attack, this is why there are several attempts to thwart the SAT-Attack, like SARLock [4], Anti-SAT [5], TTLock [6], S-URSAT [7], SFLL-HD [6] and SFLL-Rem [8]. The idea is to add structures, to increase the number of necessary DIPs to unlock the circuit. To be more specific, it takes more time to unlock the circuit instead of thwarting the SAT-based attack. Therefore, the proposed techniques increase the resilience against the SAT-based attack (SAT-resilience). Early attempts at improving SAT-resilience, like SARLock and Anti-SAT, were easy to detect in the circuit structure and could easily be removed. Attacks based on analyzing the circuit's structural properties,




(a) Basic concept of a potential LL-cell

(b) Adapted miter structure to handle activation signals

Fig. 2: Integration of potential LL-cells in the formal optimization process

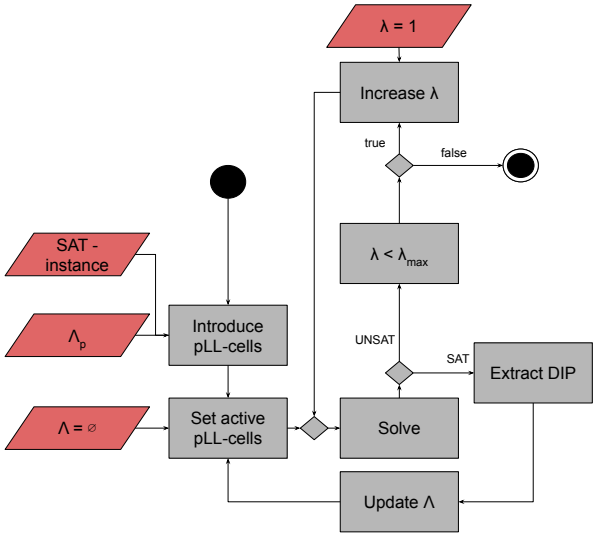



Fig. 3: Basic algorithm of the proposed method

for example, to remove or bypass encrypted signals are called structural attacks. A further type of structural attack is based on machine learning techniques like OMLA [24]. Such an attack reveals the correct key if the obfuscation of the correct key (just) depends on the logic synthesis. However, using the inherent camouflaging of PLGs like S-URSAT [7] obfuscates the correct key on a layout level, making it impossible to derive the correct key from the structure. Despite considering structural attacks in modern LL mechanisms, like SFLL-HD or SFLL-Rem, recent combinations of functional and structural attacks, like Valkyrie [10], have been able to detect the encryption logic and break these LL techniques. The structural analysis detects the artificially added encryption signal to either remove or bypass the signal. Afterward, an unlocked circuit or a circuit with low resilience against functional analysis, e.g., SAT-based attacks, is yielded.

## III. Increasing SAT-Resilience of Logic Locking Mechanisms using Formal Methods

This work proposes a novel technique to improve the SAT-resilence, while thwarting structural attacks. The approach optimizes the placement of LL-cells, while the main requirement for an LL-cell is about being configurable by a single key input.

All LL-cells with sufficient encryption and protection have to be determined. An effective LL-cell has to provide resilience against state-of-the-art attacks and address the resulting encryption. The main target of a protection mechanism is the encryption of the functional behavior if an incorrect key is applied. Nevertheless, high resilience against SAT-based attacks usually have a negative correlation to the encryption of the LL mechanism. A famous example is SARLock which only encrypts one of all possible stimuli for each incorrect key. Protection mechanisms like SARLock or similar approaches do not provide sufficient encryption for proper IP protection [25]. In conclusion, the introduced LL-cells should provide a trade-off between SAT resilience and logic encryption. Since the analysis of the structural footprint has proven a major threat against modern LL techniques, a LL-cell needs to be resistant to structural attacks. However, most attack scenarios consider the key inputs of an encrypted circuit as given, making it difficult to hide the protection mechanism since it can be detected by following the key signals. As a result, the detection of the LL structure should not unveil an encryption signal, which can be removed or bypassed to unlock the circuit. Therefore, the proposed placement method orchestrates a replacement-based LL mechanism using PLGs. Certainly, the specific placement of the LL-cells influences the encryption and SAT resilience.

The introduced LL-cells replace the CMOS gates of the original circuit with PLGs. Configuring the PLGs with the correct key inputs will yield the behavior of the original CMOS gate. By replacing the original gates, the PLGs are an inherent part of the circuit without an artificially added encryption signal that can be removed or bypassed. The LL-cells improve the SAT-resilience, by encrypting the circuit and adding keys to the search space. However, despite the low overhead of PLGs [7], a large number of key inputs significantly increases the production costs of the circuit. Therefore, it is necessary to define a maximum number of introduced LL-cells $\lambda_{max}$. Previous research like [6] considered a key size up to 256 as a reasonable number.

The proposed approach determines an SAT-resilient LL-cell placement given a maximum number of allowed key

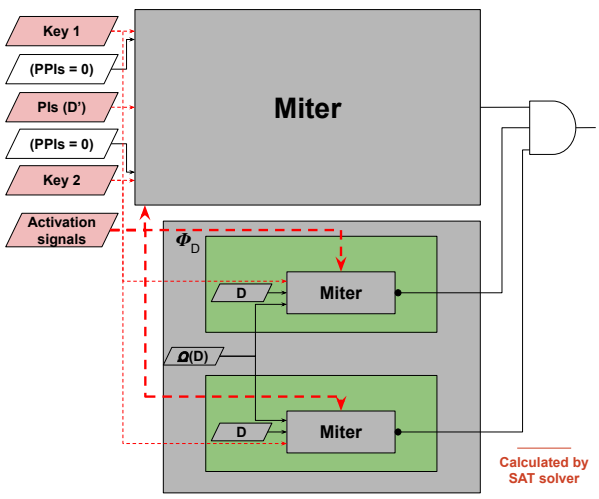



Fig. 4: Adaption of an SAT-based attack to optimize SAT-resilience

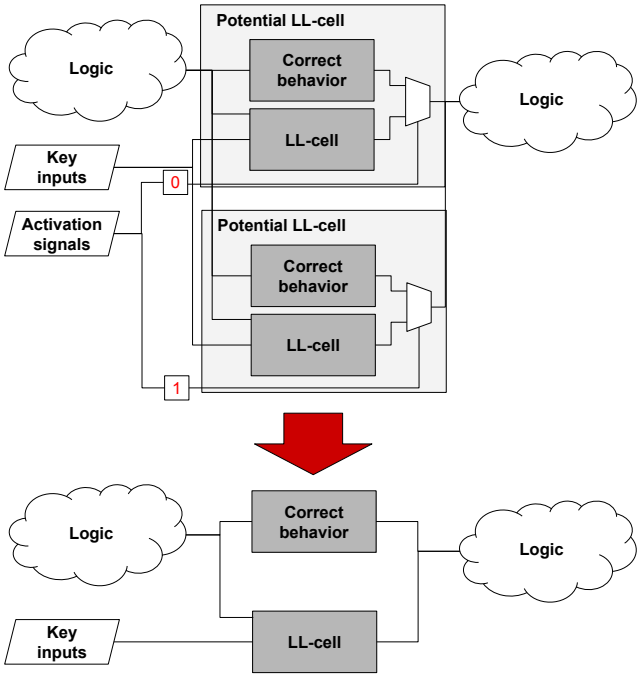



Fig. 5: Basic concept of removing activation signals yielding an optimized logic locking placement

inputs. The basic concept of the proposed method shown in Figure 3 is described in the following. First, a large set of deactivated potential LL-cells (pLL-cells) $\Lambda_p$ is introduced in the circuit. Figure 2a shows the concept of a pLL-cell. Each pLL-cell can be enabled with an activation signal, which will yield the encrypted behavior of the LL-cell. If the pLL-cell is deactivated, the correct functional behavior is applied. The algorithm is detecting a final LL-cell placement $\Lambda_f$ with $\Lambda_f \subset \Lambda_p$ and $|\Lambda_f| \leq \lambda_{max}$. To cover sequential circuits, the circuit to encrypt, including the pLL-cells, is unrolled for a predefined number of clock cycles. Afterward, the unrolled instance is copied, and similar to the SAT-based attack, a miter is constructed. The primary inputs, *Pseudo Primary Inputs* (PPIs) and activation signals of the pLL-cells are kept constant over all clock cycles and shared for both unrolled instances of the miter. Figure 2b shows an overview of the resulting miter structure. Next, the miter is transferred into an SAT instance which is solved to determine a DIP.

Initially, all introduced pLL-cells are deactivated in the unrolled instances. To limit the number of activated pLL-cells $|\Lambda|$, the SAT-instance is enhanced by $\Phi_x$, as shown in Equation 1.

$$\Phi_x = |\Lambda| \leq \lambda \tag{1}$$

$\Phi_x$ ensures that the number currently activated pLL-cells $|\Lambda|$ is lower than the number of allowed activated pLL-cells $\lambda$. $\lambda$ is initialized with $\lambda = 1$ and iteratively increased until a DIP can be determined. Afterward, the activated pLL-cells $\Lambda$ that yield the calculated DIP are added to the permanently active pLL-cells. Next, similar to the SAT-based attack, keys that behave equivalently on the DIP given the currently activated LL-cells are excluded from search space. However, contrary to the SAT-based attack, the activation signals need to be addressed during the optimization process. Figure 4 visualizes the resulting adapted DIP calculation. First, a miter is constructed using the DIP $D$ and the corresponding oracle output $\Omega(D)$. The resulting miter is added to the SAT instance sharing all activation signals calculated in $\Lambda$. Therefore, the keys excluded by a calculated DIP automatically adapt to newly calculated activation signals in $\Lambda$. Each newly introduced LL-cell provides at least one DIP that is not covered when the previously activated LL-cells have been unlocked. In particular, the minimum number of required DIPs is iteratively increased. The algorithm terminates if no more DIP can be detected while $\lambda \leq \lambda_{max}$, yielding the final placement $\Lambda_f$. In the end, the optimized placement $\Lambda_f$ is introduced in the circuit as the final protection mechanism. Figure 5 shows the detailed conversion to the final protection mechanism. More precisely, each activated pLL-cell in $\Lambda_f$ is replaced with its corresponding LL-cell, while the remaining pLL-cells are removed.

The proposed approach allows determining an SAT-resilient LL-cell placement given a large number of arbitrary pLL-cells. The LL-cells used in this work are an inherent part of the circuit and, hence, can not be removed or bypassed by structural attacks. Therefore, the proposed technique yields a protection mechanism combining resistance against structural attacks and SAT-resilience is detected.

## IV. Experimental Evaluation

This section describes the experimental evaluation of the proposed formal placement method and compares the obtained results to a random placement method.

All experiments have been conducted on an AMD 4750U with 40GB system memory. The proposed method is implemented using yices2 in a C++ environment. For evaluation, sequential circuits of the ITC'99 benchmark [11] are used. All considered circuits are unrolled for five clock cycles, which

TABLE I: Comparison of Random Placement with SAT-Attack-based placement (max. 100 LL-cells out of max. 800 pLL-cells, 5 unrolling cycles, no silent data corruption)

| Circuit | Proposed method | | | | Random Placement | | | |
|---|---|---|---|---|---|---|---|---|
| | HD | #LL-cells | #DIPs | Unlocking time [s] | HD | #LL-cells | #DIPs | Unlocking time [s] |
| b10 | 0.185 | 47 | 151 | 3388 | 0.181 | 47 | 11 | 7 |
| b11 | 0.482 | 20 | 11 | 6 | 0.116 | 20 | 2 | 3 |
| b12 | 0.247 | 24 | 4 | 73 | 0.000 | 24 | 0 | - |
| b13 | 0.171 | 28 | 13 | 5 | 0.270 | 28 | 3 | 3 |
| b14 | 0.043 | 61 | 63 | 543 | 0.008 | 61 | 5 | 168 |
| b15 | 0.021 | 28 | 5 | 793 | 0.014 | 28 | 2 | 748 |
| b17 | 0.079 | 8 | 2 | 369 | 0.000 | 8 | 0 | - |
| b19 | 0.547 | 8 | 2 | 1,904 | 0.000 | 8 | 0 | - |
| b20 | 0.119 | 63 | 189(OOM) | 13,455 (failed) | 0.061 | 63 | 3 | 216 |
| b21 | 0.100 | 50 | 56 | 905 | 0.069 | 50 | 3 | 263 |
| b22 | 0.316 | 62 | 130(OOM) | 6,039 (failed) | 0.070 | 62 | 4 | 243 |

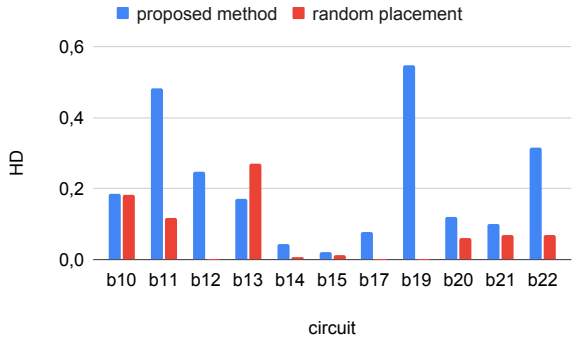



Fig. 6: Comparison of the resulting Hamming Distance

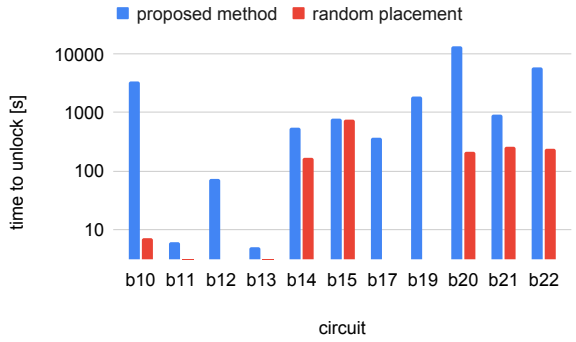



Fig. 7: Comparison of required unlocking time

has been proven an appropriate parameter to cover most of the functional behavior [26]. The proposed method initializes the state elements with 0, and the stimuli are kept constant over all five clock cycles. Therefore, only reachable states are considered. In this work, the LL-cells consist of a single PLG, resulting in a low impact on the encryption per LL-cell. However, it has proven a sufficient LL-cell for evaluating the benefits of the formal placement approach. In this evaluation an initial placement of $|\Lambda_p| = 800$ is considered, for each of the considered circuits. Therefore, 800 of the *NOR*, *NAND*, *XOR*, and *XNOR* gates are randomly replaced by pLL-cells. Consequently, each circuit contains 800 activation signals. The predefined limit of LL-cells in the final placement is $\lambda_{max} = 100$ resulting in $\sum_{x=1}^{100}(\binom{800}{x})$ possible placements of the LL-cells, which is considered a sufficiently large search-space. Furthermore, by allowing up to 100 key inputs, a large key space is obtained that can provide a non-trivial attack scenario, depending on the placement. For comparison, a random placement with the same number of active LL-cells among the 800 pLL-cells is evaluated.

Table I shows the detailed results of the proposed approach and random placement. The finally introduced LL-cells, the *Hamming Distance* (HD) [14], the number of DIPs to unlock the circuit, and the time to unlock the circuit with an SAT-based attack are shown for comparison. The HD is calculated using 10,000 randomly chosen stimuli and key combinations.

Figure 6 visualizes the resulting HD, which is used to evaluate the resulting encryption and, hence the quality of the resulting security. Optimal protection is achieved if no information about the correct output can be derived. Therefore, in [14], optimal protection is described as an average encryption of 50% incorrect output behavior. More precisely, the HD-based assessment in [14] considers an HD of 0.5 as optimal encryption. Increasing SAT-resilience usually lowers the level of encryption and vice versa since many stimuli result in correct output behavior to artificially avoid powerful DIPs. However, the encryption of the circuit is the main target of LL, to avoid malicious misuse. The proposed method is designed to increase SAT-resilience; however, it increases the encryption by adding LL-cells with DIPs. In particular, the iterative approach adds LL-cells with encrypting behavior. Furthermore, contrary to SARLock [25], the LL-cells are not artificially designed to only encrypt a single of the possible inputs. Hence the resulting encryption is applied to several stimuli and key combinations. The random approach is using randomly selected LL-cells that do not necessarily include DIPs. Therefore, it can be the case that no encryption is applied by the randomly selected LL-cells like in the case of the b12, b17, and b19.The results show that the HD of the proposed method is closer to 0.5 for each of the considered benchmark circuits. The only exception is the b13. Specific circuits like the b11 (b19) achieve an HD that with 0.482 (0.547) that converges to the target value 0.5.

Regarding the SAT-resilience, the proposed formal placement method greatly improves the number of necessary DIPs to unlock the circuit. After optimizing the placement, the circuits with the lowest SAT-resilience are b15, b17, and b19. Therefore,

there is no placement for these circuits yielding a sufficient SAT-resilience within the 800 randomly placed pLL-cells. LL-cells consisting of more PLGs can help to improve the results due to a higher impact on the behavior. Considering the b15, the number of necessary DIPs to unlock the circuit can be increased by a factor of 2.5. Furthermore, the proposed method successfully encrypts the functional behavior of the b17 and b19, while the random placement does not influence the output behavior. The highest increase in the number of necessary DIPs is achieved in the b20 and b22. The proposed approach can improve the number of necessary DIPs for the b20 by a factor of 63 and the execution time of the SAT-based attack by a factor of 62. However, both the b20 and b22 remain locked after the SAT-based attack. In particular, the iterative copying of the unrolled circuit during the SAT-based attack leads to an *Out-Of-Memory* (OOM) exception. Therefore, more time and memory than available in this evaluation is necessary to unlock the circuit. Figure 7 visualizes the time to unlock the given protected circuit using a logarithmic scaled axis. The greatest improvement regarding the time to unlock is conducted by the b10. It takes 484 times longer to unlock the b10 compared when Using the proposed formal approach over the random placement. On average the number of DIPs is increased by a factor of 19 and the time to unlock is increased by 72X.

Circuits like the b11 or b19 show that a high level of encryption usually lowers the SAT-resilience. Nevertheless, the b10, b20, b21, and b22 provide a promising trade-off between encryption, SAT-resilience, and resistance against structural attacks. All the beforementioned achieve an HD$\geq$0.1, yielding sufficient encryption of the circuit behavior when compared to SAT-resilient LL that encrypts one stimulus per incorrect key. For example, the b22 has 32 primary inputs and yields an HD of 0.119 using the proposed approach. Encrypting only a single stimulus per incorrect key obtains an HD of $\frac{1}{2^{32}}$. Furthermore, the time to unlock the circuit with the SAT-based attack is increased by at least 3.4X.

In summary, the proposed method improves the SAT-resilience for an LL-cell placement and outperforms a random placement. The approach paves the way to stop structural attacks while significantly improving SAT-resilience. The proposed method facilitates high protection against SAT-based and structural attacks as required by nowadays applications. Additionally, malicious intentions like intellectual property piracy are avoided due to high encryption.

## V. Conclusion

This paper presented an automated framework to increase the SAT-resilience for an arbitrary type of LL-cells, by improving the placement. The LL-cell used in this work is based on an RFET-based replacement technique to yield resistance against structural attacks and low hardware overhead. The evaluation showed that the proposed method clearly outperforms a random placement regarding the resulting SAT-resilience.

Future work will focus on increasing the output corruption of the LL-cells to improve the encryption on circuits like the b15.

## References


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

[2] 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, no. 2, pp. 410–424, 2015.

[3] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2015, pp. 137–143.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] D. Sirone and P. Subramanyan, "Functional Analysis Attacks on Logic Locking," in *Design, Automation and Test in Europe*, 2019, pp. 936–939.

[10] 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.

[11] F. Corno, M. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design & Test of Computers*, vol. 17, no. 3, pp. 44–53, 2000.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] J. B. Wendt and M. Potkonjak, "Hardware obfuscation using PUF-based logic," in *IEEE/ACM International Conference on Computer-Aided Design*, 2014, pp. 270–271.

[18] S. M. Plaza and I. L. Markov, "Solving the third-shift problem in IC piracy with test-aware logic locking," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 961–971, 2015.

[19] Y.-W. Lee and N. A. Touba, "Improving logic obfuscation via logic cone analysis," in *Latin American Test Symposium*, 2015, pp. 1–6.

[20] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Design, Automation and Test in Europe*, 2008, pp. 1069—1074.

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

[22] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault analysis-based logic encryption," *IEEE Transaction on Comp.*, vol. 64, no. 2, pp. 410–424, 2015.

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

[24] L. Alrahis, S. Patnaik, M. Shafique, and O. Sinanoglu, "OMLA: An Oracle-Less Machine Learning-Based Attack on Logic Locking," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1602–1606, 2022.

[25] Y. Shen and H. Zhou, "Double DIP: Re-Evaluating Security of Logic Encryption Algorithms," in *Great Lakes Symposium on VLSI*. New York, NY, USA: Association for Computing Machinery, 2017, p. 179–184.

[26] 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.