Enhancing Resilience against Sequential Attacks on Logic Locking using Evolutionary Strategies

Marcel Merten* Mohammed E. Djeridane* Muhammad Hassan*†

Niladri Bhattacharjee‡ Jens Trommer‡ Thomas Mikolajick‡§ Rolf Drechsler*†

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

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

‡NaMLab gGmbH,
01187 Dresden, Germany

§TU Dresden,
01187 Dresden, Germany

***Abstract*—Today, the manufacturing of Integrated Circuits (ICs) is highly distributed over various foundries, yielding long and untrustworthy supply chains. Therefore, severe security concerns about threats like intellectual property theft arise. Logic Locking (LL) is one well-known technique to protect a given design by introducing a secret key. Recent research increased the protection of LL mechanisms on sequential circuits by blocking access to the scan chain. However, state-of-the-art sequential attacks unlock the protection mechanism within a reasonable time resulting in a serious security threat. This work proposes a novel approach to improve resilience against sequential attacks. In particular, an Evolutionary Strategy (ES) is established to optimize the LL placement and improve the protection of the secret key. The experimental evaluation proves that the proposed hardening significantly increases the protection against sequential attacks.**

I. Introduction

Nowadays, distributed manufacturing allows designers to access advanced technology nodes without having their own semiconductor foundries. However, the globalization of chip manufacturing has become a major security challenge. Recently, protection mechanisms like camouflaging or Logic Locking (LL) techniques have been developed to avoid Intellectual Property (IP) theft. In particular, sequential LL approaches like DisOrc [1] gained interest due to the capability to prevent modern oracle-based attacks like the SAT attack [2] while providing a high output corruption.

Sequential LL focuses on the protection of sequential designs with a random LL placement by blocking the scan chain. Due to the blocked scan chain access, the attacker is limited to a Crippled Oracle Attack with No Scan Observe (COA-NSO). However, sequential attacks like RANE [3] can unlock the beforementioned protection mechanisms by observing the output behavior of multiple clock cycles. Therefore, it is mandatory to be aware of sequential attacks when introducing a LL mechanism.

This work proposes a novel optimization-based approach to enhance resilience against sequential attacks using a replacement-based sequential LL mechanism. Compared to the random key gate placements like in DisOrc, the proposed placement optimization significantly reduces the number of unlocked key bits by sequential attacks. In particular, an Evolutionary Strategy (ES) is developed to avoid key leakage beginning from the reset state of the circuit and, hence increase the protection against sequential attacks. Various experiments have been executed using the ITC'99 benchmark set [4]. The results show that compared to a random placement strategy, the proposed placement method reduces unlocked key bits, yielding superior protection against sequential attacks.

This work was financially supported by the German Federal Ministry of Education and Research BMBF under the framework of VE-CirroStrato (16ME0213) 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.

II. Enhancing Resilience against Sequential Attacks using Evolutionary Strategies

A. Evolutionary Strategies

This section introduces ESs as they are required for the proposed approach to insert an optimized LL placement. An ES simulates the natural evolution process to solve an optimization problem. Each ES consists of a population of individuals containing solutions for an optimization problem. An individual I can be represented as a triple I = (Gi, Zi, Fi) with Gi equals the genotype encoding the solution of the optimization problem, Zi holds additional side information about the addressed domain to improve the behavior of the ES, and Fi represents the fitness of i. The genotype is reflected by a set of genes, which are the substitutable atomic parts concerning the chosen encoding. Typically, an ES implements a mutation operator, to optimize the resulting genotype. The mutation operator randomly changes the genotypes yielding new genes that improve genetic diversity. Every individual i is evaluated by a fitness function $\Omega(i)$ to assess the quality of the given solution. The individuals are passed to a selection operator that removes individuals. The truncation selection is a frequently used selection operator removing individuals with the worst fitness values. Finally, the remaining population is reproduced again to replace the removed individuals if the termination condition, e.g., a certain fitness value, is not met. Note that every reproduction cycle of the ES is called generation.

B. Proposed approach

In this work, an ES is developed to optimize the LL cell placement and avoid key leakage in sequential LL mechanisms. Sequential attacks like [3] rely on starting from a known reset state to unroll the circuit until it can be unlocked. As a result, sequential attacks can become impractical, depending on the necessary unrolling steps to fully unlock the circuit. Therefore, the ES reduces the output corruption within the first clock cycles from the reset state to increase the number of necessary unrolling cycles to unlock the circuit. The proposed ES is defined by a population of individuals I_j. Each individual I_j is encoded by a genotype G_j which represents a subset of potentially active LL cells. During each generation, the mutation operator is applied to explore the search space by evaluating new combinations of active LL cells. Each individual I_j within the population performs a mutation $Mut^{\theta}(I_j)$ to determine a new individual I_k that is added to the population. The mutation operator replaces one key gate within G_j with a new key gate. In each generation of the ES, the individuals are evaluated with the fitness function Ω^{λ} defined in Equation (1) to calculate the corresponding fitness value F_j. To increase the effectiveness of the fitness calculation, the genotype G_j is distributed into equally sized partitions $\varphi \in Z_j$. During calculating the fitness F_j, the partitions defined in Z_j are evaluated individually. Since the mutation only effects a single key gate and no recombination operator is used, the fitness only has to be calculated for the partition of the effected key gate to calculate the fitness of a new individual. To calculate the fitness, the CuA is unrolled over a predefined number of clock cycles and integrated into a miter structure. The initial state λ of the unrolled Circuit under Attack (CuA) is randomly chosen from a set of reachable states Λ. The fitness is calculated over the sum of the collected corrupting keys that result in incorrect output behavior within a predefined number observed clock cycles, substracted by all possible key combinations within the evaluated partition.

$$\Omega^{\lambda}(I_j) = \sum_{\varphi \in Z_j} (2^{|\varphi|} - \#\text{corrupting keys}^{\lambda}(\varphi)) \qquad (1)$$

After calculating the fitness of each individual, the population is pruned by the selection operator. Therefore the individuals I_j in the population are sorted by their corresponding fitness values F_j. Afterward, 50% of the individuals with the lowest fitness values are removed from the population. After the selection operator is applied, the generation ends and a new generation starts with the mutation of new individuals. After evaluating a predefined number of generations, the algorithm stops and the LL placement represented by the individual with the highest fitness is used for the protection mechanism.

III. Experimental evaluation

This section evaluates the proposed optimization based placement strategy and compares it to a random placement method.

All experiments have been conducted on an AMD 4750U with 40GB system memory. The proposed approach is implemented in a C++ environment using minisat. To evaluate the protection of the placement strategy, sequential circuits of the ITC'99 benchmark [4] are used.

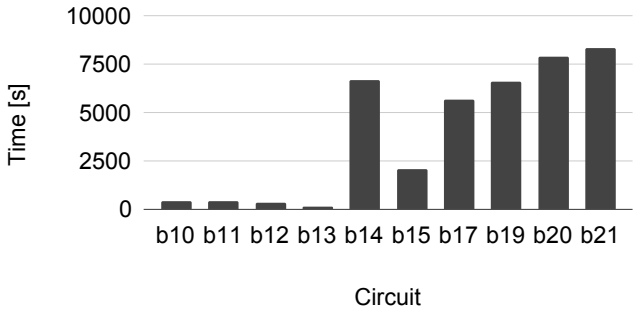

Fig. 1: Average hardening times of the proposed approach

A. Experimental Setup

First, the proposed ES is applied and the determined LL placement represented by the best individual is introduced as a protection mechanism. Afterward, a random placement strategy similar to DisOrc [1] is used for comparison. Both placement strategies are evaluated 10 times for each of the benchmark circuits to obtain a representative average and worst-case data for the considered placements. Next, the scan chain access is blocked like in DisOrc, to determine the resulting area overhead. Afterward, the placements are attacked with the sequential attack framework RANE [3]. The protection mechanisms are unlocked for up to 100 clock cycles starting from a known reset state or until reaching the predefined time-out of 12h. The placement strategies introduce a protection mechanism with a key size of $\mathcal{K} = 64$ key bits which is considered sufficiently large to represent the percent of unlocked key bits after attacking with a sequential attack. To efficiently collect the corrupting keys, the partition size is set to $S = 4$. The population size of the ES is set to 50 individuals to provide genetic diversity. During the fitness calculation, the CuA is unrolled for five clock cycles, to cover the majority of the functional behavior of the ITC'99 benchmark circuits [4]. The ES is executed for 10,000 generations before the algorithm terminates.

B. Experimental Results

Since replacement-based LL using CMOS requires additional gates to configure the circuit, the area overhead can be reduced using polymorphic reconfigurable cells with RFETs [5] instead. The reconfigurable logic gates are dynamically configured by a control signal to select between two functionalities. Dynamic XOR/XNOR and NAND/NOR cells show a 52% and 157% area overhead as compared to the simple XOR and NAND cells that they replace [6]. Using these cells within the circuits b10 and b13 the overhead including the blocked scan chain can be reduced from 181% to 162% and from 153% to 129%, respectively. The majority of the area overhead is caused by the additional logic to block the scan chain. For the larger designs b14, b15, b17, 19, b20, and 21, the total area overhead is <6%. For all circuits, the use of RFETs reduces the area overhead by about 15.55% on average.

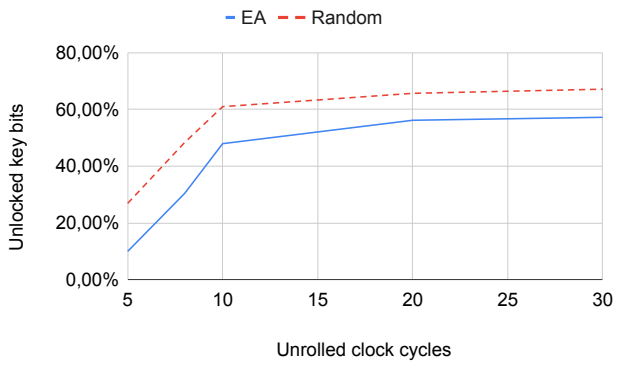

Fig. 2: Average unlocked key bits of the circuits b10 to b14 in percent

Circuit	Unlocked key bits [%]			
	Random		Proposed ES	
	time[s]	unlocked key bits	time[s]	unlocked key bits
b10	55	95.31	59	90.63
b11	1406	100.00	1517	100.00
b12	13305	32.81	18539	20.31
b13	TO	-	TO	-
b14	6118	97.41	6315	86.38
b15	TO	-	TO	-
b17	TO	-	TO	-
b19	TO	-	TO	-
b20	TO	-	TO	-
b21	TO	-	TO	-

TABLE I: RANE applied on the proposed placement (boundary 100, time-out 12h)

Figure 1 visualizes the average hardening times of the ES. The hardening times are mainly affected by the formal method used in Ω. However, with a maximum of 8341s for the b21, reasonable hardening times have been achieved.

Table I shows the average unlocked key bits for attacking the protected circuits with a boundary of 100 clock cycles. The majority of the circuits, e.g., b13, b14, b15, b17, b19, b20, and b21 hit the time-out before the attack finishes. For small circuits like the b11, a large part of the sequential behavior is covered within the first 100 clock cycles. Therefore, the sequential attack can unlock the key despite the blocked scan chain. For circuits with a higher sequential depth a larger number of introduced key bits can be protected. Considering a random placement for the b12, only 32.81% of the key bits are unlocked. However, only 20.31% key bits are unlocked on average when applying the proposed placement.

To evaluate the unlocking behavior of a sequential attack over time, the average unlocked key bits for up to 30 clock cycles are shown in Figure 2. Given the considered circuits, b10, b11, b12, b13, and b14 have been successfully attacked with a boundary of 30 clock cycles before the time-out is reached. Independent from the selected boundary, the number of protected key bits of the proposed placement is significantly higher than with random placement techniques. After 10 clock cycles, on average 61% of the key bits are unlocked given a random placement, while the proposed approach has an average of 58% unlocked key bits after 30 clock cycles.

Additionally, the figure shows that the majority of key bits are

Circuit	Unlocked key bits [%]			
	Average case		Worst case	
	Random	Proposed ES	Random	Proposed ES
b10	77.08	53.12	82.81	57.81
b11	18.77	3.65	34.38	6.25
b12	10.42	9.38	14.06	9.375
b13	28.65	15.63	35.94	18.75
b14	89.45	82.42	100.00	90.63
b15	21.88	8.60	56.25	10.94
b17	23.45	19.93	65.63	20.31
b20	95.31	83.06	100.00	87.50
b21	97.66	79.69	100.00	84.38

TABLE II: Comparison of Random Placement with proposed placement with 64 LL-cells (after unlocking circuit for the first 8 cycles from reset state)

unlocked during the first clock cycles. Therefore, the sequential attack is assessed on the first clock cycles in the following. Table II evaluates the threat of a sequential attack for the first 8 clock cycles. The results clearly show that the average number and the worst case of unlocked key bits can be improved using the proposed method. Regarding the worst case of the random placement, the b14, b20, and b21 are fully unlocked by the attack. However, with the proposed placement only about 85%-91% of the key bits can be unlocked. In the case of other circuits with a higher sequential depth like the b17, the worst case number of unlocked key bits within the first 8 clock cycles is reduced by 45.32% from 65.63% using random LL placements to 20.31% using LL placements determined by the proposed ES. Therefore, the proposed algorithm significantly improves the protection for all considered circuits.

IV. Conclusion

In conclusion, we developed a specialized ES providing an optimization-based placement strategy for key gates to increase resilience against sequential attacks. The proposed approach significantly increased the protection of the secret key on the considered circuits of the ITC'99 benchmarks [4]. Therefore, less information can be obtained by applying sequential attacks. Future work will also consider combinations with other sequential protection mechanisms.

References

[1] N. Limaye, E. Kalligeros, N. Karousos, I. G. Karybali, and O. Sinanoglu, "Thwarting all logic locking attacks: Dishonest oracle with truly random logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 9, pp. 1740–1753, 2021.

[2] 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.

[3] S. Roshanisefat, H. Mardani Kamali, H. Homayoun, and A. Sasan, "RANE: An Open-Source Formal De-obfuscation Attack for Reverse Engineering of Logic Encrypted Circuits," *Great Lakes Symposium on VLSI*, 2021.

[4] 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.

[5] 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, 2022.

[6] N. Bhattacharjee, V. Havel, N. Kavand, J. Quijada, A. Kumar, T. Mikolajick, and J. Trommer, "Dynamic reconfigurable security cells based on emerging devices integrable in fdsoi technology," *Design, Automation and Test in Europe*, pp. 1–6, 2024.