Breaking Analog Biasing Locking Techniques via Re-Synthesis

Julian Leonhard, Mohamed Elshamy, Marie-Minerve Louërat, Haralampos-G. Stratigopoulos

Breaking Analog Biasing Locking Techniques via Re-Synthesis

Julian Leonhard
Julian.Leonhard@lip6.fr
Sorbonne Université, CNRS, LIP6
Paris, France

Mohamed Elshamy
Mohamed.Elshamy@lip6.fr
Sorbonne Université, CNRS, LIP6
Paris, France

Marie-Minerve Louërat
Marie-Minerve.Louerat@lip6.fr
Sorbonne Université, CNRS, LIP6
Paris, France

Haralampos-G. Stratigopoulos
Haralampos.Stratigopoulos@lip6.fr
Sorbonne Université, CNRS, LIP6
Paris, France

ABSTRACT

We demonstrate an attack to break all analog circuit locking techniques that act upon the biasing of the circuit. The attack is based on re-synthesizing the biasing circuits and requires only the use of an optimization algorithm. It is generally applicable to any analog circuit class. For the attacker the method requires no in-depth understanding or analysis of the circuit. The attack is demonstrated on a bias-locked Low-Dropout (LDO) regulator. As the underlying optimization algorithm we employ a Genetic Algorithm (GA).

KEYWORDS

Hardware security and trust, IP/IC piracy, locking, analog circuit synthesis.

ACM Reference Format:
Julian Leonhard, Mohamed Elshamy, Marie-Minerve Louërat, and Haralampos-G. Stratigopoulos. 2021. Breaking Analog Biasing Locking Techniques via Re-Synthesis. In *26th Asia and South Pacific Design Automation Conference (ASPDAC '21), January 18–21, 2021, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3394885.3431603

1 INTRODUCTION

Intellectual property (IP) piracy has become a major threat in today's globalized semiconductor supply chain where design and manufacturing tasks are performed by many different and potentially untrusted parties. Integrated circuits (ICs) are exposed to risks such as cloning and overproduction resulting in counterfeit ICs. Cloning can be performed by (a) a System-on-Chip (SoC) integrator who obtains an IP and reuses it for other designs without remunerating again the IP owner; (b) a foundry that receives the blueprint, e.g. GDS-II file, of the IC; (c) an end-user via reverse-engineering [1]. Overproduction refers to a foundry producing ICs beyond the number agreed on the contract with the IC owner and selling them illegitimately. From the IP/IC owner perspective, cloning and overproduction result in know-how and financial losses. From the user perspective, counterfeit ICs are often of lower quality, thus endangering the application wherein they are deployed.

For digital ICs, IP/IC piracy counter-measures are being explored in research for over a decade now, with some having matured enough to be on the way of industry commercialization [2, 3]. For analog ICs, the solution space lacks largely behind with the first IP/IC piracy counter-measures published recently [4–14].

The most common defense is locking which consists in inserting a lock into the design that is controlled with a key. If an incorrect key is applied, then the functionality of the circuit is corrupted, i.e. one or more performances lie outside their specification range. The key is typically in the form of a digital key composed of k key-bits. The key size k should be large enough to circumvent brute-force attacks where the attacker randomly applies keys until finding a key that makes the circuit pass its specifications. The most common approach for locking analog ICs is biasing locking where the key acts upon the DC operating point [4–7].

Typically, with the appearance of new counter-measures, researchers aim at exploring adapted counter-attacks that challenge and oftentimes even easily break those counter-measures. In the digital domain, this back and forth between a novel defense method being established and a new attack to break it has already seen numerous iterations. This effect has lead to steady improvements in the domain of digital security. For example, the first logic locking technique for digital ICs was based on inserting key-gates into the design [15] and was later broken by the Boolean Satisfiability (SAT)-based attack that is capable of recovering the correct key with very reasonable effort [16]. Since then many new defenses have been proposed that resist the SAT-based attack, but these defenses were also compromised by other attack types. In the analog domain, the first attack was proposed recently and targets breaking biasing locking techniques [17].

In this paper, we propose an alternative attack for breaking biasing locking techniques. Compared to the attack in [17], the proposed attack is generally applicable to any analog circuit class and alleviates significantly the assumptions made regarding the capabilities of the attacker. It is inspired from analog circuit synthesis and design exploration methods [18–20]. The underlying idea is to remove the locked biasing circuit and re-synthesize it so as to recover the circuit's intent performance trade-off. The attacker will only need to rely on an optimization algorithm to complete the attack and does not need to be knowledgeable in analog circuit design. In our implementation, we use a Genetic Algorithm (GA),

ASPDAC '21, January 18–21, 2021, Tokyo, Japan

ACM ISBN 978-1-4503-7999-1/21/01…$15.00
https://doi.org/10.1145/3394885.3431603

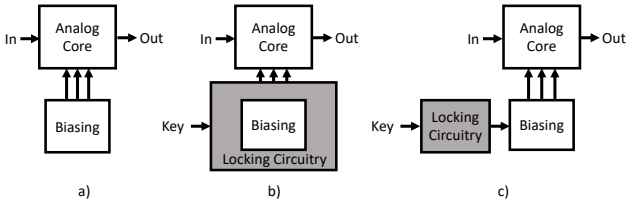

Figure 1: Analog biasing locking techniques and their different approaches.

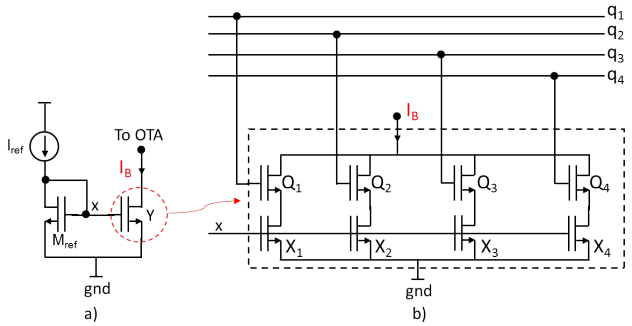

Figure 2: Locking a current mirror [6].

in particular the NSGA-II algorithm [21]. The attacker can find online a free and open-source Matlab implementation of the NSGA-II algorithm. The proposed attack is demonstrated on a low-dropout (LDO) regulator protected with biasing locking.

The rest of the paper is structured as follows. In Section 2, we review biasing locking techniques. In Section 3, we review the prior attack in [17]. In Section 4, we present the proposed attack. In Section 5, we compare the prior and proposed attacks. In Section 6, we discuss the implementation of the GA. In Section 7, we present the results on our case study. Section 8 concludes the paper.

2 ANALOG BIASING LOCKING TECHNIQUES

Analog biasing locking aims at securing the biases that set the DC operating point of the circuit, as shown in Fig. 1(a). The basic idea is to redesign the biasing circuit such that it presents some form of programmability, i.e. the resultant bias depends on the key input. There are two types of approaches, namely expanding the biasing circuit, as shown in Fig. 1(b) [5, 6], and designing a standalone block that outputs the desired bias, as shown in Fig. 1(c) [4, 7].

More specifically, in [6], it is shown how to redesign a current mirror to embed the lock mechanism. A locked current mirror with 4 key-bits q_i, $i = 1, \cdots, 4$, is illustrated in Fig. 2. The mirror transistor Y of the non-locked current mirror in Fig. 2(a) is replaced with the structure shown in Fig. 2(b) composed of 4 branches. Each branch contains a mirror transistor X_j and a switch transistor Q_j controlled by key-bit q_j. The resultant bias current I_B depends on which branches are "on" and the geometry of the mirror transistors of the "on" branches. In [5], it is proposed to replace transistor Y with parallel-connected transistors whose gate voltages are controlled by the key-bits. The key controls which transistors are "on" such that the aggregate width of the "on" transistors equals the width of the original obfuscated transistor Y. In [4], it is proposed to replace the biasing circuit with a lock mechanism based on a memristor crossbar, where the key-bits control the programming of the memristors. In [7], it is proposed to replace the biasing circuit with an on-chip neural network that receives as input an analog key in the form of DC voltages and produces at its output the desired biases. The neural network is trained to implement a delta function at the correct key, that is, any invalid key will produce incorrect biases.

Analog biasing locking is an attractive defense for several reasons: (a) it is generally applicable to all analog circuits; (b) an incorrect key will have a dramatic impact on the performance trade-off of the circuit; (c) the lock mechanism is added at the most outside layer of the circuit, thus locking is dissociated from the design of the core circuit and does not incur any performance penalties.

3 PRIOR ATTACK

The attack proposed in [17] is based on Satisfiability Modulo Theory (SMT) and is shown to break the biasing locking techniques proposed in [4–6]. We refer to it as SMT-based attack.

The threat model assumes that the attacker has access to the netlist of the locked circuit, but does not know the valid key. The attacker is also in possession of the Process Design Kit (PDK) of the technology and of the circuit data-sheet which specifies the performances and their specifications. The attacker will need to develop test benches for measuring the performances and perform circuit simulations. The most tedious and difficult aspect of the attack is that the attacker will need to write several circuit equations as will be described next. To solve these equations, the attacker will need to have access to an SMT-solver.

In particular, in a circuit with D locked biases, for the i-th locked bias b_i, $i = 1, \cdots, D$, we write an equation:

$$y_i = \phi_i(\mathbf{q}^i), \tag{1}$$

where $\mathbf{q}^i$ is the string of key-bits, i.e. $\mathbf{q}^i = [q_1^i, \cdots, q_k^i]$. For example, in the case of a locked current mirror using the technique shown in Fig. 2 [6], y_i is the aspect ratio of transistor Y, i.e. $y_i = (W/L)_Y$, and $\phi_i(\mathbf{q}^i) = \sum_{j=1}^{k} q_j^i \, (W/L)_{X_j}$, where $(W/L)_{X_j}$ is the aspect ratio of transistor X_j and $q_j^i \in \{0, 1\}$.

Next, we write an equation to express the relationship between the bias b_i and y_i:

$$b_i = \psi_i(y_i). \tag{2}$$

For example, referring to Fig. 2, we have $b_i = y_i^{'} \cdot I_{ref}$, where $y_i^{'}$ is the normalized aspect ratio of transistor Y with respect to transistor M_{ref}.

In addition, based on the m performances $\mathbf{p} = [p_1, \cdots, p_m]$ in the data-sheet, we write m equations linking each performance p_j to several biases b_i:

$$p_j = \theta_j(\mathbf{b}), \tag{3}$$

where $\mathbf{b} = [b_1, \cdots, b_D]$.

Thereafter, an SMT-solver is used to find a combination of keys $\mathbf{q} = [\mathbf{q}^1, \cdots, \mathbf{q}^D]$ that satisfies the combined equations:

$$p_j = \theta_j([\psi_1(\phi_1(\mathbf{q}^1)), \cdots, \psi_D(\phi_D(\mathbf{q}^D))]). \tag{4}$$

The SMT-based attack presents the following difficulties which limit its practicality: (a) deriving the functions θ_j is hardly possible for system-level performances. For example, considering an Analog-to-Digital Converter (ADC), main performances include Signal-to-Noise Ratio (SNR), Differential Non-Linearity (DNL), Integral Non-Linearity (INL), etc. An analog circuit's biases have a quantifiable impact on its DC operating point but not an easily

quantifiable impact on such complex system-level performances that require transient or AC analysis and that arise due to non-idealities, higher order effects, and noise. For such complex circuits, the data-sheet does not list block-level performances where the SMT-based attack could apply since those are irrelevant for the end-user; (b) functions ϕ_i, ψ_i, and θ_j can only be derived assuming simplified transistor model equations, i.e. the Spice level 1 model (aka square-law model). This introduces large imprecision and, thereby, one needs to consider margins on the bias. Due to these margins, the attack may result in a large set of possible keys, all satisfying Eq. (4). With the correct key ideally being within this set, the attacker is left with the task of launching a brute-force attack, simulating the circuit using this set of keys, in order to single out the valid key. If the key search space is large and the circuit has long simulation times, the brute-force attack turns out to be very time-consuming; (c) deriving the functions θ_j requires a very high expertise by the attacker. In other words, this attack makes very strong assumptions about the capabilities of the attacker.

4 PROPOSED ATTACK

To remedy the difficulties of the SMT-based attack we propose a novel alternative attack that leverages analog circuit synthesis.

The underlying observation is that to break biasing locking techniques it suffices to search for the correct biases instead of the key, which is arguably a much easier problem since typically there are few biases that additionally only operate within a limited range of values. Unlike the SMT-based attack that aims at unlocking the biasing circuit by extracting and applying the correct key, the proposed attack first removes the locked biasing circuit and replaces it with a "fresh" non-locked biasing circuit that is sized accordingly so as to produce the desired biases.

In our threat model, similar to the SMT-based attack, we assume that the attacker has access to the circuit netlist and PDK, knows the target performances and their specifications, and has developed test benches for measuring the performances. However, unlike the SMT-based attack which requires writing circuit equations, the proposed attack treats the circuit as a black-box. The attacker will only need to run an optimization algorithm with comprehensive decision variables and objectives.

The proposed attack has 2 versions that are described in detail below. Their high-level descriptions are illustrated in Fig. 3. In a first step common to both versions, the attacker identifies the locked biasing circuits by tracing the key-bits.

Version (a): The attacker replaces the locked biasing circuits with original non-locked biasing circuits. For example, in the case of a locked current mirror shown in Fig. 2, the attacker will reinstate the original schematic of the locked current mirror, i.e. remove the multi-branch structure shown in the right-hand side and replace it with a single transistor Y. However, the desired biases are unknown, i.e. bias current I_B in Fig. 2, and, thereby, the sizing of the components of the biasing circuits, i.e. the sizing of transistor Y in Fig. 2, are unknown. The attacker will run a multi-objective optimization algorithm to synthesize, i.e. size, the biasing circuits in the context of the complete circuit with the aim of satisfying all performance specifications. The decision variables are the values of the components in the biasing circuits, i.e. transistor aspect ratios, resistor values, etc. For example, in the case of a locked current

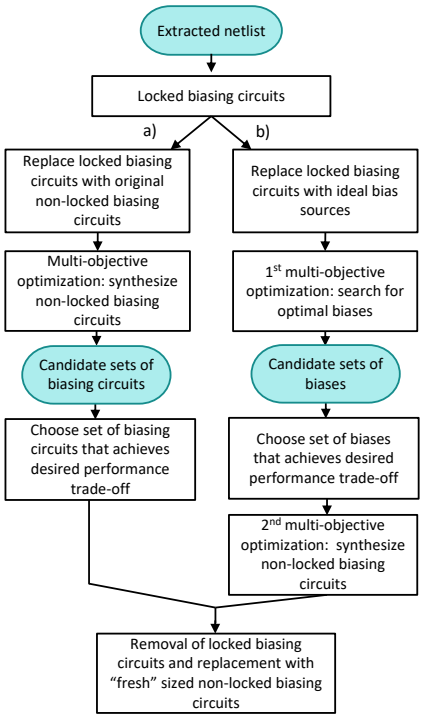

Figure 3: Flowchart of the 2 versions of the proposed attack.

mirror shown in Fig. 2, there is a single decision variable, i.e. the width W of transistor Y since the length L is left untouched by locking. The objective function is a metric of performance trade-off for the circuit, i.e. the Euclidean distance from the specification boundaries measured with the $\|\cdot\|_2$ norm. During optimization, for every visited set of decision variables, the optimizer generates the corresponding circuit netlist and queries the circuit simulator to measure the performances using the test benches and compute the objective function. The optimization algorithm will return a Pareto front with p Pareto optimal solutions of performance trade-offs. A solution is considered Pareto optimal or non-dominated when no performance can be improved further without degrading any other performance. Each Pareto solution is produced by a specific set of sized biasing circuits. Then, the attacker has the freedom to choose a Pareto optimal solution that best meets the performance trade-off goal. It is noteworthy that the biases found may be different from the original obfuscated biases, but will still achieve similar or even better performance trade-offs. Finally, the attacker proceeds with the removal of the locked biasing circuits and their replacement with the corresponding sized biasing circuits resulting from optimization.

Version (b): The attacker removes the locked biasing circuits and replaces them with ideal bias sources. In this case, a first optimization is run using the biases as decision variables. The attacker will choose a Pareto optimal solution that meets the performance trade-off goal, and, in a second optimization step, the attacker will synthesize original non-locked biasing circuits independently of the core circuit such that they produce the biases resulting from the first optimization. Finally, the attacker will combine the sized non-locked biasing circuits with the core circuit and will run a confirmatory simulation so as to verify the performance trade-off.

Table 1: Requirements for analog bias locking attacks

	Proposed attack	SMT attack [17]
Attack type	Removal	Key recovery
Shared requirements	PDK, data-sheet, netlist, testbenches, circuit simulator	
Additional requirements	Optimization algorithm, e.g. GA	Equations ϕ, ψ, θ SMT-solver
Attacker expertise	Low	High

So far we have not made any mention to the underlying optimization algorithm. In fact, any optimization algorithm can be used. In our implementation we use a GA that will be described in more detail in Section 6.

5 COMPARISON

Conceptually, the difference between the proposed attack and the SMT-based attack is that the proposed attack is a removal attack aiming at automatically redesigning the locked biasing circuits, whereas the SMT-based attack aims at extracting the valid key.

A direct comparison regarding the required capabilities of the attacker for the two attacks is shown in Table 1. First, the SMT-based attack requires deriving circuit equations, which is hardly possible for many circuit classes, as discussed in Section 3. In contrast, the proposed attack requires only a common optimization algorithm and is generally applicable to any circuit class. Second, deriving circuit equations is a complex task requiring high analog design expertise, whereas with the proposed attack the attacker does not need to have any analog design expertise. In fact, the circuit can be handled as a black-box. Therefore, the proposed attack can be implemented by a "weak" attacker and, thus, it is considerably more powerful than the SMT-based attack.

6 GENETIC ALGORITHM

In our implementation, we employ a GA for performing heuristic multi-objective optimization. GAs are global optimization methods inspired by natural selection in nature. Mechanisms such as survival of the fittest, mutation and crossover are applied to a population to create descendant populations with ever-improving performances. Performances of individuals are evaluated through objective functions (aka fitness functions). In particular, we employ a Matlab implementation of the NSGA-II GA [21]. It is a free and open-source code with comprehensive parameters to be set, thus it can be readily used by any adversary to perform successfully the attack. The NSGA-II GA is configured as follows:

- A decision variable, i.e. component values such as transistor width for version (a) of the attack and bias values for version (b) of the attack, is represented with a gene. A gene is real-coded with real-valued floating point vectors. The concatenation of all genes represents the chromosome corresponding to a biasing circuit netlist for version (a) of the attack and biases for version (b) of the attack.
- We constrain the optimization by defining loose boundaries in the decision variables space so as to avoid unrealistic values, i.e. gigantic transistor widths and negative biases.
- We define loose boundaries for the performances set at a value equal to the specification multiplied by a factor of two. For chromosomes that have performances outside this range we penalize their fitness function such that they are disregarded from the next generation. In this way, the search is constrained within meaningful performance trade-offs.
- The selection function uses a binary tournament, where two randomly chosen candidates of the parent generation compete. The individual dominating the other, i.e. with a better rank, wins the tournament. When candidates with equal ranks are competing the crowding distance is decisive, by preferring solutions that are in less crowded regions. We are computing the crowding distance in the decision variable space which ensures a diverse population.
- The number of elite individuals to be reused in a new generation is controlled in order to preserve a diverse population. We limited the number of individuals on the Pareto front, i.e. of rank 1, to 35 % of the population.
- The chosen crossover operator uses line recombination, meaning that the offspring lies on a random point on the line between its two parents. 80 % of a new generation - excluding elite children - are generated using the crossover operator.
- We chose a Gaussian mutation operator, i.e. a random number chosen from a Gaussian distribution is added to each gene of a parent. 20 % of a new generation - excluding elite children - is generated using the mutation operator.
- Other settings are as follows: population size 50, number of generations 100.

7 RESULTS

Our case-study is an LDO regulator whose role is to provide stable supply voltages to sensitive circuits and is therefore widely applied in all sorts of ICs. The LDO regulator must compensate for variations in e.g. load current, temperature or the global supply voltage. This makes an LDO regulator an interesting circuit to lock, especially in the context of a SoC with many sub-circuits that require stable supply voltages within small margins to operate correctly. With an incorrect key, a locked LDO regulator will provide out-of-spec supply voltages and, thereby, disrupt functionality of parts or the entire SoC.

We designed an LDO regulator in a 65nm CMOS technology using the free and open-source OCEANE tool [22]. The block-level schematic of the LDO regulator is shown in Fig. 4. The transistor-level schematics of the error amplifier circuit, implemented with an operational transconductance amplifier (OTA), and the bandgap voltage reference circuit are shown in Figs. 5 and 6, respectively. Going down in the LDO regulator hierarchy, the bandgap voltage reference circuit includes an internal amplifier implemented with a self-biased OTA (SOTA) whose transistor-level schematic is shown in Fig. 7. The error amplifier in Fig. 5 and internal amplifier in Fig. 7 require proper biasing provided through current mirrors.

The LDO regulator is locked via locking these two current mirrors using the technique shown in Fig. 2 [6]. In Figs. 5 and 7 we highlight the obfuscated mirror transistor, i.e. transistor Y referring to Fig. 2.

The main performances of the LDO regulator that we consider include: (a) output voltage dependence on temperature variations

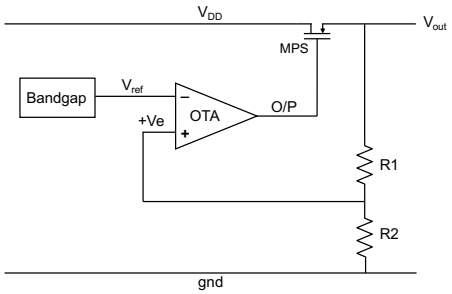

Figure 4: LDO regulator block-level schematic.

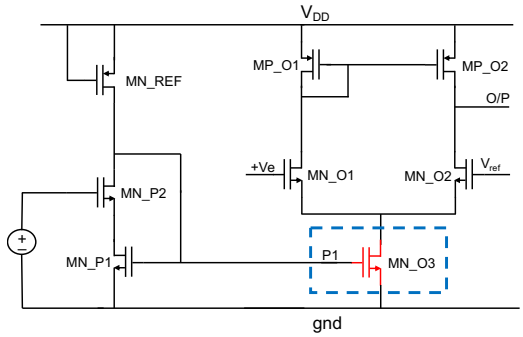

Figure 5: Error amplifier circuit inside the LDO regulator.

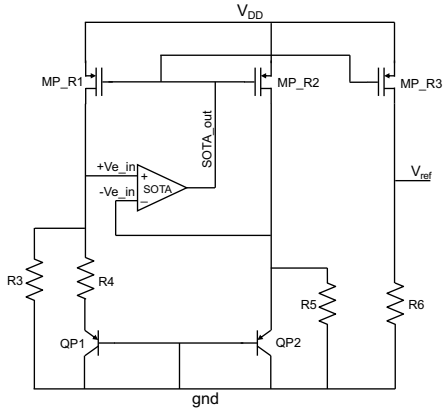

Figure 6: Bandgap voltage reference circuit inside the LDO regulator.

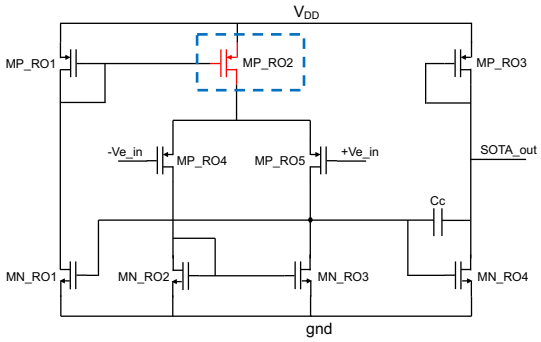

Figure 7: Internal amplifier circuit inside the bandgap voltage reference circuit.

from −55 °C to 125 °C, i.e. $\frac{\Delta V_{\text{out}}}{\Delta T}\left[\frac{\text{mV}}{\text{°C}}\right]$; (b) output voltage dependence on supply voltage variations from V_{DD} = 1.5 V to 3 V, i.e. $\frac{\Delta V_{\text{out}}}{\Delta V_{DD}}\left[\frac{\text{mV}}{\text{V}}\right]$; (c) output voltage offset from the nominal output voltage of 1.18 V under full and zero load current, denoted by ΔV_{out} @ 0 mA [mV] and ΔV_{out} @ 50 mA [mV], respectively; (d) regulated output voltage overshoot as a response to a sudden variation of the load current from 0 mA to 50 mA.

Table 2: Recovered LDO regulator performance trade-offs resulting from the attack.

	$\frac{\Delta V_{\text{out}}}{\Delta T}\left[\frac{\text{mV}}{\text{°C}}\right]$	$\frac{\Delta V_{\text{out}}}{\Delta V_{DD}}\left[\frac{\text{mV}}{\text{V}}\right]$	ΔV_{out} @0 mA [mV]	ΔV_{out} @50 mA [mV]	V_{out} overshoot [mV]
Nominal	8.11	33.45	1.12	3.15	47.97
Version a) Pareto sol. #					
1	5.73	34.74	0.91	4.09	35.45
2	8.69	32.95	0.98	3.11	51.20
3	7.94	35.89	0.35	3.60	38.52
4	5.58	35.17	1.14	3.85	36.57
5	6.24	35.42	0.62	3.83	36.74
Version b) Pareto sol. #					
1	10.43	33.26	0.30	3.11	51.67
2	4.83	35.00	2.05	3.64	38.06
3	12.53	36.54	3.27	3.40	41.02
4	8.99	36.05	0.99	3.53	39.27
5	9.27	34.59	0.14	3.22	45.20

Table 2 shows the nominal performances and 5 diverse Pareto solutions produced by each attack version. All performances have an upper specification. Figs. 8, 9, and 10 plot the LDO regulator characteristic measurements for the nominal design, a locked design when applying a random incorrect key, and the unlocked design using Pareto solution #2 of version (a) of the attack. As it can be seen from Table 2, both versions of the attack are capable of fully recovering the circuit functionality, resulting in good performance trade-offs compared to the nominal one. In fact, thanks to the optimization, the attack is even capable of finding improved performance trade-offs.

To visualize the Pareto front and appreciate the diversity of Pareto solutions, we run the attack by considering only two performances, namely ΔV_{out} @ 50 mA and V_{out} overshoot. The Pareto front is illustrated in Fig. 11.

A single call to the simulator to extract all performances takes approximately 5.8 seconds on an Intel Xeon E5-2640 @ 2.40 GHz with 128 GB of RAM. To perform the single optimization in version (a) of the attack and the first optimization in version (b) of the attack, we run a GA using 100 generations with a population of 50 per generation, totaling in 5000 simulator calls. The second optimization in version (b) of the attack sizes the standalone biasing circuit for which simulation time is very small. In this case, the GA terminates in less than 5 minutes. Therefore, the time of version (a) of the attack is 8.06 hours and the time of version (b) of the attack is only slightly bigger.

8 CONCLUSION

We demonstrated an attack that breaks any biasing locking technique. The attack is based on removing the locked biasing circuits and re-synthesizing them. The attack applied to a protected LDO regulator is completed in about 8 hours and recovers an LDO regulator with excellent performance trade-off. It requires only the

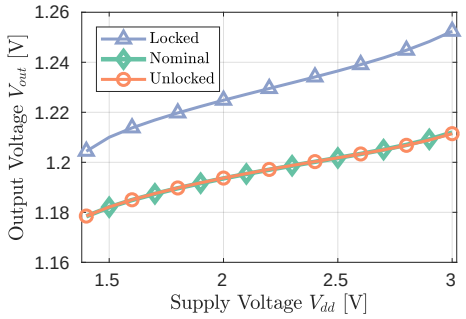

Figure 8: LDO regulator output voltage vs. supply voltage variations.

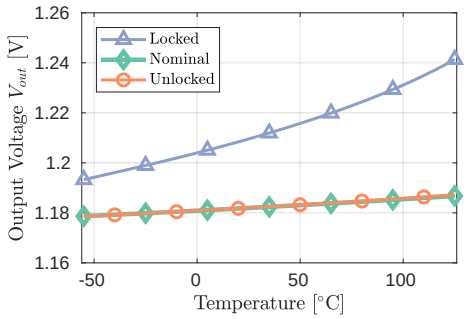

Figure 9: LDO regulator output voltage vs. temperature variations.

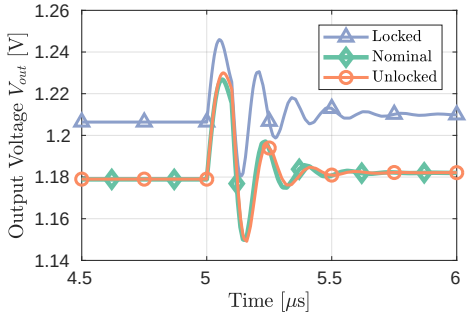

Figure 10: LDO regulator output voltage vs. time showing the response to a sudden variation of the load current.

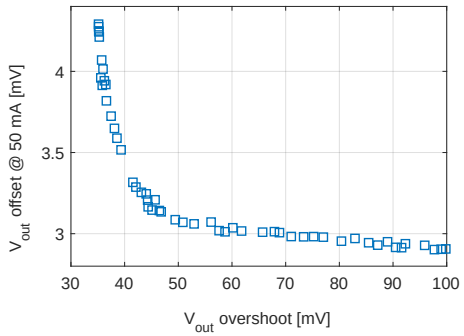

Figure 11: Pareto front showing trade-off between two optimization objectives: ΔV_{out} @ 50 mA vs. V_{out} overshoot.

use of an optimization algorithm, thus it can be performed even by "weak" attackers that have no analog design expertise. In addition, the attack is generally applicable to any analog circuit.

ACKNOWLEDGMENTS

This work has been carried out in the framework of the ANR STEALTH project with N° ANR-17-CE24-0022-01. J. Leonhard has a fellowship from the doctoral school EDITE de Paris.

REFERENCES

[1] B. Lippmann, M. Werner, N. Unverricht, A. Singla, P. Egger, A. Dübotzky, H. Gieser, M. Rasche, O. Kellermann, and H. Graeb, "Integrated flow for reverse engineering of nanoscale technologies," in *Proc. Asia and South Pacific Design Automation Conference*, p. 82–89, 2019.

[2] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[3] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP protection and supply chain security through logic obfuscation: A systematic overview," *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 6, pp. 65:1–65:36, 2019.

[4] D. H. K. Hoe, J. Rajendran, and R. Karri, "Towards secure analog designs: A secure sense amplifier using memristors," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2014.

[5] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *Proc. IEEE Latin American Test Symposium*, 2017.

[6] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting analog IC piracy via combinational locking," in *Proc. IEEE International Test Conference*, 2017.

[7] G. Volanis, Y. Lu, S. Govinda, R. Nimmalapudi, A. Antonopoulos, A. Marshall, and Y. Makris, "Analog performance locking through neural network-based biasing," in *Proc. IEEE VLSI Test Symposium*, 2019.

[8] N. G. Jayasankaran, A. S. Borbon, E. Sanchez-Sinencio, J. Hu, and J. Rajendran, "Towards provably-secure analog and mixed-signal locking against overproduction," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2018.

[9] S. G. R. Nimmalapudi, G. Volanis, Y. Lu, A. Antonopoulos, A. Marshall, and Y. Makris, "Range-controlled floating-gate transistors: A unified solution for unlocking and calibrating analog ICs," in *Proc. Design, Automation and Test in Europe Conference*, 2020.

[10] M. Elshamy, A. Sayed, M.-M. Louërat, A. Rhouni, H. Aboushady, and H.-G. Stratigopoulos, "Securing programmable analog ICs against piracy," in *Proc. Design, Automation and Test in Europe Conference*, 2020.

[11] J. Leonhard, M. Yasin, S. Turk, M. Nabeel, M.-M. Louërat, R. Chotin-Avot, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "MixLock: Securing mixed-signal circuits via logic locking," in *Proc. Design, Automation & Test in Europe Conference*, 2019.

[12] J. Leonhard, M.-M. Louërat, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "Mixed-signal hardware security using mixLock: Demonstration in an audio application," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019.

[13] A. Ash-Saki and S. Ghosh, "How multi-threshold designs can protect analog IPs," in *Proc. IEEE International Conference on Computer Design*, pp. 464–471, 2018.

[14] J. Leonhard, A. Sayed, M. Louërat, H. Aboushady, and H. Stratigopoulos, "Analog and mixed-signal IC security via sizing camouflaging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020. early access.

[15] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending piracy of integrated circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.

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

[17] N. G. Jayasankaran, A. S. Borbon, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking analog locking techniques via satisfiability modulo theories," in *Proc. IEEE International Test Conference*, 2019. Paper 9.1.

[18] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.

[19] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137 – 148, 2009.

[20] N. Lourenço, R. Martins, A. Canelas, R. Póvoa, and N. Horta, "AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation," *Integration*, vol. 55, pp. 316 – 329, 2016.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[22] J. Porte, "Outil pour la conception et l'enseignement d'électronique analogique (OCEANE)." https://www-soc.lip6.fr/equipe-cian/logiciels/oceane/. Online.