Hidden Costs of Analog Deobfuscation Attacks

Vaibhav Venugopal Rao, *Student Member, IEEE*, Kyle Juretus, *Member, IEEE*, and Ioannis Savidis, *Senior Member, IEEE*

***Abstract*— Analog obfuscation techniques to prevent intellectual property attacks have mainly evolved from digital obfuscation. Similar to digital hardware security, the considered threat models commonly assume that the attacker possesses the circuit netlist, specifications, and bias information to deobfuscate a locked analog circuit. However, when one or more pieces of information remain unavailable, there is an adverse effect on the performance of current analog attack algorithms. In this article, an analysis of the challenges and limitations of obtaining the information needed to successfully attack an analog circuit is provided. In addition, the performance of current state-of-the-art analog attack techniques is evaluated when one or more pieces of information is unavailable. The analysis of the attack on five distinct analog circuits obfuscated with key-based parameter locking is performed, premised upon the level of information possessed by the adversary. The monotonic attack (MA) returned the correct key in less than 10 h when executing a black-box attack on single stage circuits obfuscated with a 10-bit key. The key-spacing (KS) attack is 10× faster than the monotonic attack and returns 8.3× fewer candidate keys for multistage analog circuits. The satisfiability modulo theory (SMT) based attack is 224× slower than the monotonic attack and 2240× slower than the key spacing attack for an 18-bit obfuscated circuit. A genetic algorithm (GA) based attack is 121 091× slower than an monotonic attack even for a single stage analog circuit. Through analysis of the results, metrics are developed to characterize the setup and evaluation time of executing the deobfuscation attacks.**

***Index Terms*— Analog obfuscation, satisfiability (SAT), satisfiability modulo theory (SMT), secure circuit design.**

I. Introduction

TO ADDRESS the growing concern of analog hardware security and to rapidly develop methods for analog obfuscation, techniques and methodologies are borrowed from the digital domain, including split manufacturing [1], key-based performance locking [2], [3], [4], [5], [6], and the use of multithreshold transistors [7]. Different attack algorithms, including satisfiability modulo theory (SMT) based [8], [9] and optimization-based [10], have been proposed to evaluate the efficacy of techniques that obfuscate analog circuits. Some attack methodologies assume an oracle assisted model similar to that of the satisfiability (SAT) based attacks in the digital domain [11], [12], [13]. However, unlike the digital circuit domain, where the SAT attack is the de facto standard for measuring the resiliency of key-based logic locking techniques, no such standard exists in the analog circuit domain.

Current analog attack algorithms assume an ideal threat model in which an attacker possesses all circuit information except for the correct key. However, when considering an attack executed in the field, an adversary typically does not possess all circuit information. Unlike for digital circuits, where only the circuit netlist and a live integrated circuit (IC) are needed for execution of the SAT attack to determine the key obfuscating the combinational logic [11], [12], [13], attack algorithms require additional information to effectively deobfuscate a locked analog circuit. As an example, for the equation based SMT attack proposed in [8], additional information needed for the execution of the attack includes the biasing conditions of the circuit and process design kit (PDK) specific parameters such as the mobility of electrons/holes (μ), the gate oxide capacitance (C_{ox}), the thickness of the gate oxide (t_{ox}), and the threshold voltage (V_{th}). The loss of one or more sources of information adversely affects the performance of the analog attack algorithms, where, on average, the equation based SMT attack [8] returns 30× more candidate keys and requires 17× more time to execute when the attacker does not possess the required PDK information as compared to the case proposed in [8] for an attacker assumed to possess all required information. Similar trends are observed for other analog attack algorithms, including the genetic algorithm (GA) attack [10] and the monotonic response attack [9].

This article is the first to analyze the performance and applicability of analog obfuscation attacks executed on five analog benchmark circuits with varying levels of information available to the adversary. The primary contributions of this article include the following.

1) Addressing the hidden cost, number of steps, and resources needed to successfully attack an analog circuit. Unlike the threat models for obfuscated digital circuits that require only the circuit netlist and a live IC to execute the SAT attack, analog attack techniques also require information on the biasing conditions of the circuit and technology specific parameters found in the PDKs. The cost, challenges, and resources needed to obtain the required information are overlooked when considering current threat models but are considered in this article.
2) The current state-of-the-art attack techniques assume that the adversary possesses all required circuit

Manuscript received 1 February 2023; revised 23 May 2023; accepted 15 June 2023. Date of publication 20 September 2023; date of current version 24 October 2023. This work was supported in part by the Drexel Ventures Innovation Fund and in part by the National Science Foundation under Grant CNS-1751032. *(Corresponding author: Vaibhav Venugopal Rao.)*

Vaibhav Venugopal Rao and Ioannis Savidis are with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: vv85@drexel.edu; is338@drexel.edu).

Kyle Juretus is with the Department of Electrical and Computer Engineering, Villanova University, Villanova, PA 19085 USA (e-mail: kyle.juretus@villanova.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3296279.

Digital Object Identifier 10.1109/TVLSI.2023.3296279

information. However, an attack executed in the field is far from ideal, where the attacker lacks one or more sources of required information. In this article, four separate threat models are assumed that more closely mimic real-world attack scenarios, with one or more sources of needed information made progressively unavailable to the attacker.

3) Current analog attack techniques are not evaluated on multistage and multinode obfuscated analog circuits. This article, for the first time, evaluates current analog attack techniques on five distinct analog circuit topologies that include a varying number of stages, obfuscated nodes, and key sizes.
4) Novel metrics are proposed to evaluate the performance of the attack algorithms for the four real-world scenarios based on the resources and the time needed to initialize and execute an attack. The metrics provide insight on the resources needed to successfully attack the locked analog circuits and provide a means to compare analog attack algorithms across the four real-world attack scenarios considered.

This article is structured as follows. Background information on prior analog obfuscation techniques is presented in Section II. A discussion on the challenges of obtaining circuit information to successfully attack an analog circuit when assuming an ideal attack model is provided in Section III. A brief overview of the benchmark analog circuits used for evaluation of the attack algorithms is provided in Section IV. The four scenarios developed to evaluate the attack algorithms are described in Section V, while the results of the evaluation are presented in Section VI. A description of the metrics used to evaluate the performance of the analog attack techniques is provided in Section VII. A concise summary of the results and some conclusions are provided in Sections VIII and IX, respectively.

II. Prior Work on Analog Obfuscation

An overview of prior work in analog obfuscation is provided in this section. Split manufacturing was one of the first techniques proposed to prevent piracy of an analog IC [1]. With split manufacturing, the fabrication of an analog IC is divided into a front-end-of-line (FEOL) process and a back-end-of-line (BEOL) process. A FEOL process consists of transistor layers fabricated by an untrusted foundry. A BEOL process consisting of metalization layers that are used in the fabrication of both interconnects and passive components (resistors, inductors, and certain capacitors) is fabricated by a trusted foundry. As a result, an untrusted foundry fabricating the FEOL must explore a large design space to reverse engineer the complete analog IC. Since the complete IC is provided after fabrication and packaging, split manufacturing is only effective against reverse engineering from untrusted foundries during fabrication and does not prevent IP piracy from an untrusted end user. A 16-bit key-based locking mechanism that utilizes memristors was implemented to tune the body voltage of the input transistors of a sense amplifier circuit [3]. Although the proposed method is effective against an evil-maid attack [3], the practical application of the technique is limited as memristors are not readily available and the memristor fabrication process is incompatible with the standard CMOS fabrication process [14]. A method that replaces a small number of nominal V_{TH} (NVT) transistors with low V_{TH} (LVT) and/or high V_{TH} (HVT) transistors while maintaining the target performance specifications of the circuit has been proposed in [7]. The limitation of implementing multithreshold voltages as a security feature is that the technique is only applicable to large analog circuits, as the threshold voltages of the devices of a smaller circuit are relatively easy to determine using a brute-force attack. In addition, advances in imaging techniques and methodologies provide a means to determine the dopant concentration of the silicon, which reveals the threshold voltage of the transistors [15].

Additional key-based locking mechanisms for analog circuits have been proposed [2], [4], [5], [6], [16], [17], where locking circuitry is inserted into the IC. A locked design produces a correct output only on the application of the correct key. In [2], [4], [16], and [17], the physical dimensions of the transistors used to set the optimal biasing conditions of the circuit are obfuscated. In [4], combinational logic locking is applied to the current mirrors of the circuit, where transistors of different sizes are used to mask the current gains. The logic locking technique proposed in [5] and [6] obfuscates the digital blocks of a mixed-signal IC through the implementation of stripped-functionality logic locking [18].

III. Assumptions in Threat Models for Attack Techniques

To quantify the robustness of an attack algorithm, an adversary is assumed to possess a certain level of information pertaining to the obfuscated circuit under attack. However, obtaining the information when assuming practical considerations is challenging, time consuming, and expensive. In this section, an analysis of the challenges in obtaining the required information is provided.

A. Netlist Extraction

For the threat model of an untrusted foundry, netlist extraction is a straightforward procedure of converting a GDSII file to a netlist. However, if the attacker is an untrusted end user, the process of extracting the circuit netlist is costly and expensive and requires dedicated facilities [19]. To completely reverse engineer an IC, high-quality images of both active layers and metal layers are needed. The reverse engineering process typically consists of five primary steps: depackaging, delayering, imaging, annotation, and extraction [19]. Depackaging requires either use of corrosive chemicals or dedicated mechanical equipment. In the delayering process, each of the metal layers is removed and scanned until the substrate is reached. Since each layer varies in thickness and type, the delayering methodology for each layer differs. In addition, the delayering process requires the use of dedicated facilities with strict climate control and the use of corrosive chemicals, which adds time and expense. The delayered IC is imaged using a scanning electron microscope (SEM) to extract a precise representation of each metal and active layer. The cost of

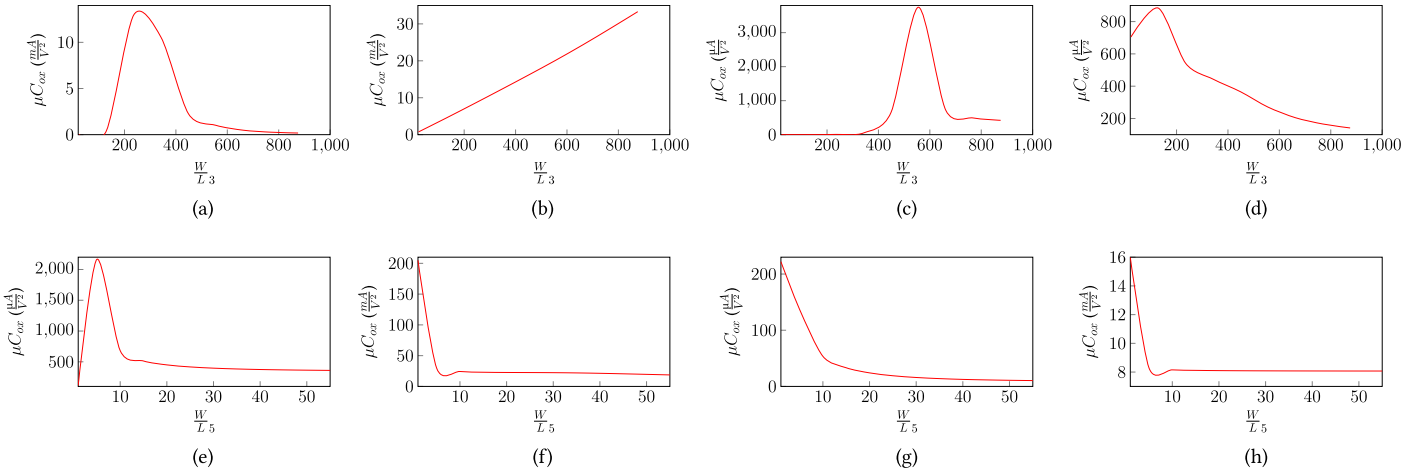

Fig. 1. Analysis of the variation in μC_{ox} of the critical transistors of (a)–(d) the single stage VGA shown in Fig. 2(a) and (e)–(h) the op-amp shown in Fig. 3 when obfuscating the transistor dimensions ($\frac{W}{L}$) of, respectively, M_3 and M_5. Characterization of (a) μC_{ox} variation of M_1 (VGA), (b) μC_{ox} variation of M_3 (VGA), (c) μC_{ox} variation of M_5 (VGA), (d) μC_{ox} variation of M_7 (VGA), (e) μC_{ox} variation of M_1 (op-amp), (f) μC_{ox} variation of M_5 (op-amp), (g) μC_{ox} variation of M_7 (op-amp), and (h) μC_{ox} variation of M_8 (op-amp).

SEM imaging increases with advanced technology nodes [20]. Once the IC is delayered and imaged, the transistors and metal layers are identified and aligned to determine the connections between devices. The process of annotation requires dedicated reverse engineering software tools that align and help visualize all layers of the IC. Depending on the accuracy of the annotation, the netlist is automatically generated.

The most advanced imaging tools utilize ptychographic X-ray laminography, which requires around 30 h to scan 0.3 mm^2 of a side with 500-nm resolution [21]. In addition, to scan a 40-μm^2 area of a circuit, the laminography process requires approximately 60 h to complete when resolving 19-nm features. State-of-the art laminography techniques are only able to reverse engineer interconnects and are not mature enough to reverse engineer transistors [21]. Extrapolating the data provided in [21], reverse engineering the interconnect of a 3-mm^2 analog receiver using X-ray laminography requires about 12.5 days for 500-nm resolution and 187.5 days for 19-nm resolution. A typical cost of reverse engineering an IC at the block level ranges from \$10 000 to \$50 000, while the extraction of the full circuit netlist is estimated to cost upward of a quarter million dollars [22]. In addition, the tools utilized for annotation are typically capable of identifying digital gates and interconnects implemented with standard place and route tools. However, additional steps are required to identify custom designed circuits and routing typically seen in analog and mixed signal ICs. Therefore, the described steps highlight the complexity, cost, resources, and time needed to completely extract the netlist from a packaged analog IC.

B. Extraction of PDK Information

When considering an untrusted foundry, the PDK information is assumed readily available to the adversary. However, when considering an untrusted end user, the attacker must gain access to the PDK to extract technology specific parameters including the charge mobility (μ), gate oxide thickness (t_{ox}), and threshold voltage (V_{th}). The technology specific parameters do not remain constant, as shown in Fig. 1, where β, given by $\mu C_{ox}(\frac{W}{L})$, varies by as much as 1500$\times$ and 4000$\times$ when considering the entire range of obfuscated transistor dimensions of an op-amp and variable gain amplifier (VGA), respectively. Therefore, assuming typical values for technology specific parameters does not guarantee the return of the correct key in the list of candidate keys. In addition, allowing for a range of values for technology specific parameters increases the likelihood of returning the correct key at the cost of increased complexity of the SMT attack, increased execution time, and an increase in the number of returned keys that the attacker must then brute force.

C. Extraction of Bias Voltages and Currents

Irrespective of the threat model, determining the bias voltages and/or currents is challenging as neither the GDSII nor the datasheets include the biasing information of the internal node(s). When considering an untrusted foundry, the attacker is only able to estimate the range of the bias voltages and currents by analyzing the biasing circuitry. The bias voltage for most analog circuits is typically between the threshold voltage (V_{th}) and the power supply voltage (VDD). For an untrusted end user, the biasing voltages and/or currents are determined in one of two ways: 1) by netlist extraction through reverse engineering and SPICE simulation or 2) a probing attack on the depackaged IC. The extraction of the netlist through reverse engineering is a complex, expensive, time consuming, and destructive process that results in the loss of the bias voltages and/or currents of the circuit. Therefore, similar to an untrusted foundry, the attacker must determine the ranges of the bias voltages and currents by analyzing the biasing circuitry.

A probing attack is an invasive method used to characterize the functionality of an IC by depackaging and electrically probing internal nodes [23]. The steps involved in a probing attack include: 1) decapsulation, 2) reverse engineering, and 3) wire location and probing. The attacker must reverse engineer the entire IC to analyze and determine the locations to probe. Since the original IC used for reverse engineering

is no longer functional, the probing attack is performed on a separate IC. For the second IC, the attacker must once again perform decapsulation, where the IC packaging is removed to expose the die. An attacker must ensure that the die is not damaged during the decapsulation process [23]. Once the IC is prepared and nodes for probing are identified, the next step is to determine the absolute coordinates of the point(s) to mill, which requires a precise kinematic mount fiducial markers. State-of-the-art focused ion beam (FIB) is then utilized to mill the hole(s) that expose the target wiring [23]. Once a target wire is exposed, probes must be placed in contact with the target interconnect without triggering, if implemented, any alarm signals produced by active or passive shielding techniques [23]. With contact to the target wire, the bias voltage and/or current is extracted using a probe station, which potentially leads to an erroneous bias point due to the impedance of the probes. The described steps highlight the complexity, cost, and resources needed to extract the bias voltages and/or currents of an analog circuit.

D. Generating Circuit Models

In this article, to reduce the time needed to determine the key of a locked analog IC, the attacks are performed on a model of a locked analog circuit. Performing an attack on a physical circuit typically involves preparing the IC sample for attack, inputting a key, extracting voltage/current information through probing, determining the circuit response, and inputting the circuit response into an attack algorithm, which is mostly manual and time intensive. Therefore, circuit models are essential to efficiently attack a locked analog IC, where the accuracy of a circuit model directly affects the performance of an attack algorithm.

Generating a SPICE netlist of an analog IC is part of the reverse engineering process. However, the accuracy of the extracted SPICE netlist is dependent on the accuracy of the utilized imaging, annotation, and extraction tools, as discussed in Section III-A. The generation of circuit equations utilized by the SMT attack requires reverse engineering the locked analog IC, identifying the different subblocks of the IC (amplifier, mixer, phase-locked loops, filters, and so on), determining the specific topology of a given subblock (i.e., for an amplifier, determining whether the amplifier is single-ended, differential, an operational amplifier, an instrumentation amplifier, or a cascode amplifier), determining the equations that best model each of the subblocks, and correctly combining the equations of all the subblocks utilized by the analog attack algorithm. The efficacy of the equation-based models, therefore, depends on the accuracy of the reverse engineering process and the equations that model each of the subblocks. As technology scales, the accuracy of the equations describing the behavior of the circuit diverges from the simulated and experimental behavior of the circuit, which, therefore, results in less accurate circuit models for attack [24]. In addition, the technology specific information of a particular process node and the biasing conditions of the circuit also affect the accuracy of the developed models. To compensate for the variation in the performance parameters resulting from inaccurate circuit models, as compared to the performance of an active analog IC, the attack algorithms allow for a range of acceptable performance values.

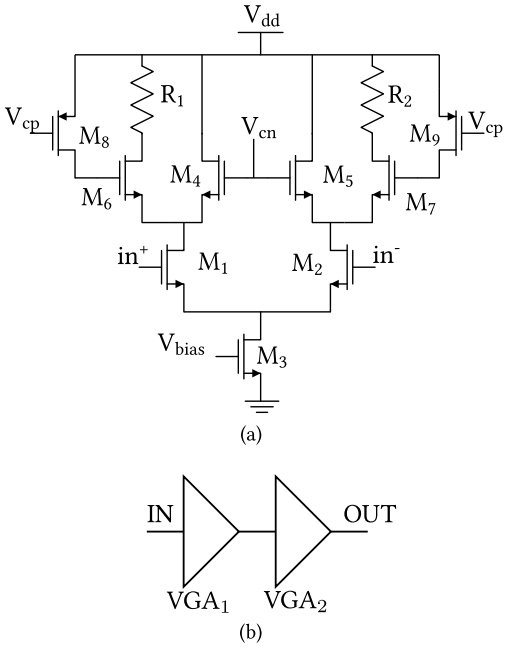

Fig. 2. Circuit schematic of (a) single stage VGA [25] and (b) two stage VGA. Each circuit is secured with key-based parameter obfuscation, where the single stage VGA is obfuscated by a 10-bit key and the two stage VGA is obfuscated by a 12-bit key.

IV. Circuit Implementations

The four analog circuits that are secured and then used for evaluation of the analog obfuscation attacks are discussed in this section.

A. Variable Gain Amplifier

The circuit schematic of a single stage VGA is shown in Fig. 2(a). Two topologies of a VGA are secured utilizing key-based parameter obfuscation, one consisting of a single stage and the other consisting of two stages. Transistor M_3 of the single stage VGA is obfuscated with a 10-bit key to mask both the target output gain of 8.3 dB and the target bandwidth of 100 MHz. The circuit schematic of the two stage VGA is shown in Fig. 2(b), where two single stage VGAs are serially connected. A 12-bit key is utilized to obfuscate the total combined gain of the two stage VGA, which is set to 14.34 dB for a bandwidth of 100 MHz. A 6-bit key is used to obfuscate the M_3 transistor of each stage of the two stage VGA.

B. Operational Amplifier

The circuit schematic of a two stage op-amp is shown in Fig. 3. The target specifications of the op-amp that are obfuscated include a gain of 60 dB, a gain bandwidth (GBW) of 10 MHz, and a slew rate of 10 V/μs. Both the first and second stages are secured by obfuscating, respectively, the sizes of transistors M_5 and M_8 each with a 6-bit key (12 bits total).

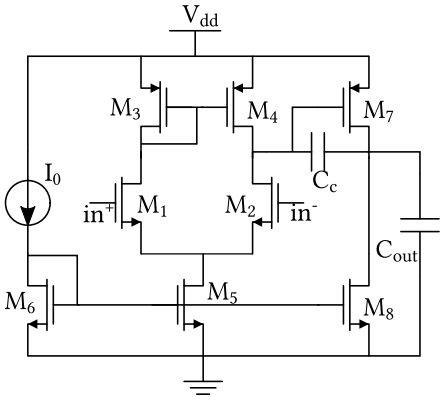

Fig. 3. Circuit schematic of a two stage operational amplifier [26].

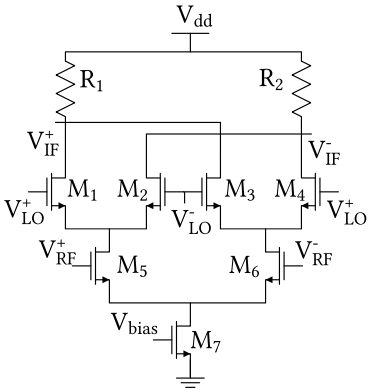

Fig. 4. Circuit schematic of a Gilbert mixer [27].

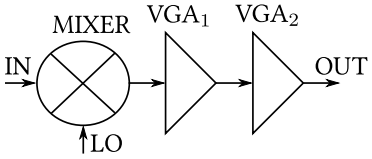

Fig. 5. Block diagram of a front-end circuit consisting of a mixer and a two stage VGA.

C. Mixer

The circuit schematic of a mixer is shown in Fig. 4. The mixer is designed to produce a 100-MHz output frequency V_{IF} for an input RF frequency V_{RF} of 2.4 GHz and a local oscillator frequency V_{LO} of 2.3 GHz. The conversion gain (CG) of the mixer is set to 3.25 dB. The circuit is secured by obfuscating the dimensions of transistor M_7 with a 10-bit key, which masks the CG of the mixer.

D. Front-End

An analog front-end circuit consisting of a mixer and a two stage VGA is designed as shown by the block-level diagram in Fig. 5. The VGAs and the mixer are designed with the same specifications as described in Sections IV-A and IV-C, respectively. Each subblock of the front-end circuit is obfuscated with a 6-bit key, which results in an overall key length of 18 bits. Transistor M_7 of the mixer shown in Fig. 4 is obfuscated to mask the conversion gain (CG), while transistor M_3 of each VGA shown in Fig. 2(a) is obfuscated to mask the overall gain A_v and center frequency F_c of the front-end. Therefore, obfuscating a transistor in each of the three subblocks of the front-end masks the overall target gain and the operating frequency of the multistage circuit.

V. Security Evaluation of Analog Attack Algorithms

In this section, four analog attack scenarios, based on the level of circuit information available to the adversary, are described.

1) Scenario I – Attacker Possesses All Information Except the Correct Key: Most analog attack algorithms are currently evaluated only for Scenario I, where the attacker possesses the obfuscated circuit netlist, all bias voltages and/or currents of the circuit, the PDK documentation, all circuit specifications, and an oracle IC. Utilizing the given information, the adversary efficiently models the obfuscated circuit and executes the attack. Attacks executed considering Scenario I typically ignore the cost and time of reverse engineering the netlist through delayering and imaging, obtaining the bias information through probing, obtaining the appropriate PDK information from the foundry, and obtaining and properly characterizing the oracle IC, which makes the attack both the most expensive and the most time consuming.

2) Scenario II – Attacker Does Not Possess PDK Information: In addition to not having the correct key, the attacker is no longer assumed to possess the exact PDK information needed to develop the electrical model of the circuit, which includes the mobility μ, the gate oxide capacitance C_{ox}, the gate oxide thickness T_{ox}, and the threshold voltage V_{th} of the devices. In addition, each electrical parameter varies with circuit conditions, which are also assumed unavailable to the attacker. However, based on the implemented circuit and basic technology information, the attacker is able to make an educated guess on the range of possible values each PDK parameter can assume. The circuit models developed under Scenario II are not as accurate and the search space of the attack algorithms is much larger than that of Scenario I, which results in an increase in the number of iterations of the attack, a longer time to execute, and a greater number of returned candidate keys that must be verified against the oracle IC.

3) Scenario III – Attacker Does Not Possess Biasing Information: For Scenario III, the attacker is assumed not to possess biasing information and PDK documentation while attempting to determine the correct key. The cost and time required to obtain the biasing information through probing is no longer considered. However, the circuit netlist and the PDK provide information on the biasing circuitry, through which the attacker determines applicable ranges of biasing voltages and/or currents.

4) Scenario IV – Attacker Only Possesses the Oracle IC: For Scenario IV, the attacker only possesses the oracle IC. The attacker is assumed capable of obtaining I/O responses from the oracle for different applied keys. The attacks executed while considering Scenario IV are both the cheapest and fastest to initialize but typically require more time to execute and return a larger number of candidate keys as compared to the other scenarios.

TABLE I

RESULTS OF EXECUTING THE EQUATION-BASED SMT ATTACK ON THE FIVE OBFUSCATED ANALOG BENCHMARK CIRCUITS DESCRIBED IN SECTION III AND FOR THE FOUR DIFFERENT SCENARIOS DESCRIBED IN SECTION IV. "*" INDICATES THAT THE SMT CONSTRAINTS ARE FORMULATED TO ACCOUNT FOR THE GAIN, GAIN BANDWIDTH, AND SLEW RATE

Scen-arios	VGA (10-bits)				Mixer (10-bits)				Op-Amp (12-bits)				2-stage VGA (12-bits)				Front-End (18-bits)			
	Toler-ance	Keys	Time	Comment	Toler-ance	Keys	Time	Comment	Toler-ance	Keys	Time	Comment	Toler-ance	Keys	Time	Comment	Toler-ance	Keys	Time	Comment
I	10%	294	2.34 s	No correct key	10%	2	0.62 s	No Correct key	10%	138	2.56 s	Correct key	10%	735	14.64 s	No correct key	10%	9735	799.97 s	No correct key
									10%*	0	0.18 s	UNSAT								
	50%	1009	6.30 s	Correct key	60%	8	0.67 s	Correct key	50%*	78	2.05 s	Correct key	50%	3969	43.60 s	Correct key	50%	42948	2818.25 s	Correct key
II	10%	295	2.51 s	No correct key	10%	607	3.86 s	Correct key	10%	4089	31.96 s	Correct key	10%	3969	51.07 s	Correct key	10%	250047	13.35 H	Correct key
									10%*	0	0.13 s	UNSAT								
	50%	1009	6.74 s	Correct key					60%*	123	3.59 s	Correct key								
III	10%	511	3.75 s	Correct key	10%	1023	3.87 s	Correct key	10%	4089	42.24 s	Correct key	10%	3969	92.63 s	Correct key	10%	250047	14.67 H	Correct key
									50%*	251	19.59 s	Correct key								
IV	Cannot Perform Attack																			

VI. PERFORMANCE OF DIFFERENT ANALOG ATTACK METHODOLOGIES

In this section, the SMT [8], GA [10], key-spacing (KS) [9], and monotonic-based [9] attacks are evaluated on obfuscated circuits for the four attack scenarios described in Section V. The evaluation of the analog attacks is performed on a server that includes two 12-core Intel Xeon E5 CPUs and 96 GB of DDR4 memory.

A. Equation-Based SMT Attack

An SMT based technique to circumvent analog obfuscation is proposed in [8]. The technique requires formulating SMT equations based on the extracted netlist and on knowledge of the analog circuit, PDK, bias voltages and currents, and circuit specifications. The SMT problem is formulated from two fundamental equations derived from the analog circuit under attack. For the formulation of the first equation, the physical locations or subblocks of the circuit that are locked and the type of locking techniques are determined. The relationship between the obfuscated parameter(s) and the biasing conditions (bias voltages and/or currents) is then formulated through analog circuit knowledge. The second equation is formulated based on the circuit performance of the oracle IC. The relationship between the analog circuit specification(s) and the biasing conditions is based on the topology and the type of analog circuit implemented. Finally, the SMT problem is formulated by equating the two derived equations, where the biasing conditions determined by setting the obfuscated parameter(s) of the analog circuit must equal the biasing conditions of the oracle IC that result in the correct analog circuit performances.

The described threat model [8] assumes that the attacker already has information on: 1) the sizes of all the transistors and passive devices of the obfuscated analog circuit except for the locked component(s) from the extracted netlist, 2) the technology-dependent parameters, including the mobility of electrons/holes (μ), gate oxide capacitance (C_{ox}), thickness of the gate oxide (t_{ox}), and the threshold voltage (V_{th}), 3) the set biasing voltages and currents, and 4) the circuit specifications from the datasheets.

The equation-based SMT attack is executed on the benchmark analog circuits described in Section IV and for the four attack scenarios presented in Section V. The results obtained from the analysis of the SMT attack are listed in Table I, where tolerance refers to the difference between the parameter specifications obtained from the model of the analog circuit and the specifications obtained from the oracle IC. The SMT algorithm returns all the candidate keys that meet the circuit specifications within the defined search space. If a key exists within the candidate keys that produces the target circuit specifications, then the key is labeled as a correct key. For the equation-based SMT attack, the adversary must determine the correct key from the list of candidate keys through brute force. The results obtained by setting the tolerance to 10%, which is the minimum tolerance used for the analysis, and the lowest tolerance for which the SMT attack returns the correct key, given as a percentage, are listed in Table I. The objective from the perspective of the attacker is to minimize the tolerance while also ensuring the presence of the correct key within the list of candidate keys. Setting a lower tolerance results in a fewer number of returned candidate keys and a faster execution time of the attack algorithm. However, the presence of the correct key is not always guaranteed, as indicated by the results listed in Table I for a tolerance of 10%. In contrast, setting a high tolerance increases the probability of finding the correct key within the list of candidate keys. However, a greater number of candidate keys are also returned, which the attacker must then brute force to determine the correct key.

The SMT attack performs exceptionally well for Scenario I, where the adversary possesses all circuit information except for the correct key and is, therefore, able to construct an accurate model of the obfuscated circuit. The SMT attack terminates within seconds and results in only a handful of candidate keys. A brute-force attack is then utilized on the candidate keys to determine the correct key. However, as the information that the attacker possesses is reduced, the search space drastically increases such that the SMT solver now has to search through ranges of PDK parameters and biasing voltages and currents. As the number of considered variables increases, the execution time and the number of returned candidate keys also increase. For the single stage VGA considered under Scenarios II and III and for a mixer considered under Scenario III, less than 1.5% of the total keys are eliminated.

For the op-amp, the SMT algorithm is constrained for two conditions: 1) (Type I) only the gain parameter is used to

TABLE II

RESULTS OF EXECUTING THE GA ATTACK ON THE FIVE OBFUSCATED ANALOG BENCHMARK CIRCUITS DESCRIBED IN SECTION III AND FOR THE FOUR DIFFERENT SCENARIOS DESCRIBED IN SECTION IV

Scenarios	VGA (10-bits)					Mixer (10-bits)					Op-Amp (12-bits)					2-Stage VGA (12-bits)					Front-End (18-bits)				
	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment
I-II	1%	35	30	15.4 H	No Correct Key	1%	35	30	15.3 H	No Correct Key	1%	65	65	65.5 H	No Correct Key	1%	65	65	65.7 H	No Correct Key	5%	90	85	Timeout	No Key
	5%	25	20	7.4 H	Correct Key	5%	20	15	4.16 H	Correct Key	5%	25	25	9.3 H	Correct Key	50%	65	65	65.7 H	No Correct Key	50%	85	85	114.3 H	No Correct Key
III - IV	10%	30	25	13.3 H	Correct Key	5%	30	35	11.2 H	Correct Key	10%	55	60	50.3 H	Correct Key	50%	65	65	65.7 H	No Correct Key	50%	85	85	114.3 H	No Correct Key

constrain the solver and 2) (Type II) gain, GBW, and slew rate are used to constrain the solver. The results of the Type II analysis are denoted with an "∗" in the tolerance column of Table I. The results of performing the equation-based SMT attack on the op-amp indicate that increasing the number of circuit specifications to constrain the formulated SMT problem reduces the number of returned candidate keys as compared to constraining the algorithm with a single circuit specification. However, an increase in the probability of UNSAT (algorithm returning no keys) is observed. The number of returned candidate keys for the Type II analysis is reduced by over 95% as compared to the Type I analysis performed when considering Scenario II, even when allowing for a high tolerance of 60% for the Type II analysis. For a standalone analog circuit (single stage), determining different circuit parameters is straightforward (from the datasheet). However, for a larger analog circuit consisting of many subblocks (i.e., the analog front-end described in this article), determining multiple circuit specifications is a challenge, as described in Section III. The results from the analysis indicate that the SMT attack is only viable for single stage analog circuits, where only one biasing node is obfuscated and the attacker is able to develop a highly accurate circuit model with tight parameter ranges to represent the oracle IC.

B. Genetic Algorithm-Based Attack

In [10], an attack is proposed, which utilizes a genetic attack to determine the correct key used to secure an analog circuit. The GA attack requires information on the circuit netlist, biasing conditions, and circuit specifications. Based on all the necessary information, the attacker sets the population size, number of generations, probability of mutation, probability of crossover, chromosome size, and the initial chromosome structure [10]. In addition, based on the target circuit performances, the attacker formulates a cost function (CF) that is minimized when executing the GA attack.

The GA attack consists of iteratively executing both the GA algorithm and simulation on the extracted netlist. The GA algorithm determines a key based on the starting chromosome. The determined key is applied as an input to the circuit netlist, where the simulator PySpice returns the characterized performance of the circuit for the given key. The simulated circuit performances are then compared to the target circuit performances for calculation of the CF. The GA algorithm tunes the chromosomes that produce the lowest cost. The steps involved in execution of the GA attack include the following.

1) Obtaining the circuit netlist through reverse engineering, and identification of the portion of the circuit that is obfuscated.
2) Mapping of the keys obfuscating the circuit to a chromosome.
3) Formulation of the CF to minimize the error between the oracle response and the modeled circuit response based on the circuit specifications obtained from the oracle IC.
4) Optimization of the search space and determination of the key that results in the lowest cost based on the specified population size and the number of generations.

For the GA attack, the CF is formulated such that an applied key is only considered a possible candidate if there is a 5% or less error between the target and simulated performance values. The probability of mutation is set to 20% and the number of generations and the population size are each initially set to five. Both the number of generations and the population size are then incremented by five until the GA attack produces the same key for three consecutive random initial chromosome values. For the analysis in this article, the GA attack described in [10] is modified to utilize Cadence and Spectre instead of PySpice for modeling, characterization, and analysis of the circuit response. The use of Cadence and Spectre greatly improves the accuracy of the results and also allows for the utilization of complex AC and noise-related simulations at the expense of increased execution time. The single key returned by the GA attack is not guaranteed to be the actual target key used to lock the analog circuit and greatly depends on the formulation of the CF. For the GA attack executed under Scenario I, all the information including biasing conditions, the PDK documentation, and the circuit specifications, is assumed to be available to the adversary. Since the attack utilizes Cadence and Spectre, the PDK parameters, including μC_{ox}, T_{ox}, and V_{th}, are already assumed available. Therefore, the results for Scenarios I and II are the same. For Scenario III, the attacker must also determine the biasing information in addition to the obfuscating key, where both unknowns are mapped to a single chromosome. The number of chromosome bits is defined by the size of the obfuscating key and the resolution of the biasing information. The algorithm inputs the chromosome into the simulator and based on the circuit response, tunes the chromosome such that the CF results in a lower error. The attacker is assumed to possess access to the key inputs and the output pins of the oracle IC, through which input–output responses are observed. Based on the responses, the chromosome values are tuned. In this article, the GA attack is evaluated on the schematics of the analog benchmark circuits, which are also considered as the oracle circuits. Therefore, the attack executed considering Scenarios III and IV is identical.

The results of executing the GA attack algorithm on the five analog benchmark circuits are listed in Table II. For

all cases, the analysis of the GA attack is performed by initially setting the maximum allowed error of the CF to 1% and both the population and generation sizes to 5. The CF is defined as the absolute value of the difference between the target circuit specification(s) and the simulated circuit specification(s) over the target circuit specification(s) as a percentage ($\frac{|\text{Target}_{\text{output}} - \text{Simulated}_{\text{output}}|}{\text{Target}_{\text{output}}} \cdot 100$). The timeout for each attack is set to five days (120 h). The population and the generations are each incremented by 5 until the five-day timeout is reached, at which point, the error in the CF is increased to 5%. The CF is increased by an additional 5% if further timeouts occur. Incrementing the maximum allowed error of the CF increases the probability of determining the correct key. The population and generations are both reset to an initial value of 5 each time the CF is incremented.

For a single stage circuit obfuscated by a 10-bit key, the product of the population size and generation size is set to approximately 2^{10}. The maximum population is set to 35 and the maximum number of generations is set to 30, which results in a total number of iterations of $35 \cdot 30 = 1050$, which is approximately equal to 2^{10}. The results indicate that setting a very low CF of 1% does not lead to the correct key for any of the benchmark circuits even after the GA attack performs an equal number of iterations to a brute-force attack. When the CF error is increased to 5%, the GA attack determines the correct key for the single stage benchmark circuits, specifically the VGA and the mixer, in less than 8 h.

For the two stage circuits, specifically the op-amp and the two stage VGA, the maximum population and the generation size are each set to 65, where the product of the two corresponds to the number of unique keys produced by a 12-bit key ($65 \cdot 65 = 4225$, which is approximately equal to 2^{12}). Similarly, for the front-end circuit, the population and the number of generations are each set to 85, as any higher value in either the population or the generation results in a timeout. Executing the GA attack on the multistage analog circuits failed to determine the correct key even after the GA attack performs an equal number of iterations to a brute-force attack. However, for a two stage obfuscated op-amp, the GA attack determined the correct key in 9.3 h, as, unlike the two stage VGA and the front-end, the op-amp response is mostly monotonic with respect to both the obfuscated transistor dimensions and the key bits and includes a smaller number of local minima. In contrast, the two stage VGA and the front-end circuit produce nonmonotonic responses and consist of many local minima and maxima. Optimization algorithms, including the GA, do not easily escape from such local minima and maxima. The GA attack, therefore, failed to determine the correct key applied to secure the two stage VGA and the front-end.

C. Key Spacing

The key spacing attack proposed in [9] leverages a vulnerability present in key-based obfuscation techniques. To avoid multiple correct keys and to ensure that an incorrect key produces a significant degradation in the performance of a secured analog circuit, an obfuscated circuit requires an exclusion zone around the target device dimensions. The attack utilizes an SMT solver to determine candidate widths with sufficient spacing from the next closest widths, where the spacing interval is set by the attacker. The attack continues until no further candidate widths are determined. The threat model considered for the KS attack is that of an adversary that possesses the circuit netlist, through which the obfuscated transistor sizes are targeted.

Algorithm 1 Key Spacing Attack

Input: Width Values $\vec{W}_v$,
Key Spacing Variance Var;
$W = \vec{W}_v * \vec{X}$;
$S_1 = W \wedge (sum(\vec{X}) > 0)$;
$candidate\_keys = []$;
while $SAT[S_i]$ **do**
  $W_a = find\_adjacent\_width(W_i)$;
  $var\_spacing = W_i * Var$;
  if $W_a - W_i >= var\_spacing$ **then**
    $candidate\_keys.append(W_i)$;
  $S_{i+1} = S_i \wedge (\vec{W}_v < \vec{W}_{v_i} \vee \vec{W}_v > \vec{W}_{v_i} + var\_spacing)$;
end
return candidate_keys;

The pseudocode for the KS attack is provided as Algorithm 1. The SMT solver is utilized to determine the active width segments of an obfuscated transistor width W based on the key $\vec{X}$, where $\vec{X} \in b[0, 1]$. The nearest width W_a greater than W is determined through execution of an optimization algorithm. If W_a is greater than the specified tolerance given by *Var*, then W is added to the list of candidate keys. The constraints applied to the model representing the circuit provided to the SMT solver are updated to ensure that the next W generated is less than W_i or greater than $W_i + var\_spacing$, where W_i is the candidate key from the previous SMT iteration and *var_spacing* is the separation distance between the given width and the next allowed width (exclusion zone). The constraint holds as any key between W_i and $W_i + var\_spacing$ does not satisfy the KS constraint. As *var_spacing* is increased, the number of keys eliminated per iteration increases. The process is repeated until there are no additional satisfying widths.

The SMT analysis is performed to evaluate the execution of the key spacing attack on the five analog circuits described in Section IV and for the four attack scenarios described in Section V, with results as listed in Table III. The attack is efficient in pruning the key space, where the correct key for an 18-bit obfuscated front-end circuit was determined in less than 2 s. The user defined key spacing variance (*Var*) is the primary parameter to consider when executing the KS attack algorithm, which defines the spacing between two adjacent keys. If the *Var* between two keys is greater than or equal to the user defined parameter, then the keys are considered as potential candidates. Setting a larger *Var* results in the faster pruning of the key space and a smaller number of returned candidate keys but does not guarantee that the correct key is included among the candidate keys as indicated by the last row

TABLE III
RESULTS OF EXECUTING THE KS ATTACK ON THE FIVE OBFUSCATED ANALOG BENCHMARK CIRCUITS DESCRIBED IN SECTION III AND FOR THE FOUR DIFFERENT SCENARIOS DESCRIBED IN SECTION IV

Scenarios	VGA (10-bits)				Mixer (10-bits)				Op-Amp (12-bits)				2-stage VGA (12-bits)				Front-End (18-bits)			
	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment
I-III	1%	19	1.93 s	Correct Key	1%	11	3.02 s	Correct key	1%	44	0.52 s	Correct Key	1%	27	0.89 s	Correct Key	1%	36	1.25 s	Correct Key
	5%	8	1.27 s	Correct key	5%	7	1.34 s	Correct key	5%	40	0.48 s	Correct key	5%	13	0.91 s	Correct key	5%	21	1.46 s	Correct key
	25%	2	0.45 s	No Correct key	25%	2	0.81 s	No Correct key	30%	4	0.33 s	No Correct key	15%	9	0.61 s	No Correct key	15%	12	0.93 s	No Correct key
IV	Cannot Perform Attack																			

of Scenarios I–III of Table III for a *Var* of 25%. In contrast, setting a lower *Var* results in increased computational time and an increased number of candidate keys that the attacker must then brute force but guarantees the presence of the correct key among the candidate keys. Since the KS attack only requires the netlist of the extracted circuit, Scenarios I–III result in the same solutions. However, the attack is infeasible for Scenario IV, where only the oracle IC is available to the adversary.

D. Monotonic Circuit Response Attack

For the monotonic attack (MA), an adversary is assumed to have access to the input–output responses of the obfuscated circuit. The attack algorithm utilizes the monotonic relationship between the given width and the output response of the circuit, which applies to a majority of single stage analog circuits. The input–output responses utilized to develop a model for the attack are determined in one of two ways: 1) simulation on the extracted netlist using circuit simulators or 2) characterization of the circuit response of the oracle IC based on the applied key, where the adversary is assumed to possess access to the key inputs and there exists a one-to-one relation between the key inputs and the obfuscated circuit parameters.

The pseudocode for the monotonic function attack is provided as Algorithm 2. The attack begins by querying the attack model for the response of the circuit to an applied input and given key. A parameter offset is also applied to account for any process, voltage, and/or temperature (PVT) variations that alter the output of the circuit. The response generated by the attack model for a given key and input is utilized to analyze the parameter characteristics of the circuit (i.e., gain, bandwidth, and operating frequency). The algorithm checks whether the response of the circuit falls outside of the range of permitted values as compared to the response generated by the oracle, and if so, constrains the selection of the width during the next execution of the solver. If the returned response falls within the accepted range of circuit properties, then the given key is added to the list of candidate keys. The process continues until no further widths are generated. The algorithm concludes by returning the list of candidate keys.

The results of executing the MA on the five analog test circuits described in Section IV and for the four different attack scenarios described in Section V are listed in Table IV. For Scenarios I–III, where the attacker possesses the circuit netlist, the MA is performed utilizing analog circuit models constructed from generic circuit equations with an attack flow, as shown in Fig. 6(a). However, for Scenario IV, where the attacker only possesses the oracle IC, Cadence-based schematic simulations are utilized only to emulate the oracle response, as shown in Fig. 6(b), with no knowledge of the circuit topology. The results of executing the MA on the five analog circuits indicate that the attack is very effective in determining the correct key for Scenarios I–III, where generic circuit equations based on the extracted netlist are used to model the obfuscated analog circuit. Candidate keys were returned in less than 1 s when attacking the single stage circuits obfuscated with a 10-bit key, less than 10 s when attacking the two stage circuits obfuscated with a 12-bit key, and less than 1 h when attacking the three stage front-end circuit obfuscated with an 18-bit key. The relatively fast execution time of the MA is a result of ignoring both the transistor operating region and the nonlinear parameter characteristics of the devices when developing equation-based models of the circuit. Consequently, the circuit responds monotonically and the key search space is halved each iteration. The results of executing the MA for Scenario IV indicate that the technique is effective only for obfuscated single stage analog circuits, specifically the single stage VGA and the mixer. However, as the number of circuit stages increases, the circuit response becomes nonmonotonic, which results in the failure of the attack to determine the correct key of the op-amp, the two stage VGA, and the front-end, as indicated by the results listed in Table IV for Scenario IV.

Algorithm 2 Monotonic Attack

Input: Width Values $\vec{W}_v$,
Variance Var;
$W = \overrightarrow{W}_v * \overrightarrow{X}$;
$S_1 = W \wedge (sum(\vec{X}) > 0)$;
$candidate\_keys = []$;
$oracle\_response = query\_oracle()$;
$oracle\_min = oracle\_response * (1 - Var)$;
$oracle\_max = oracle\_response * (1 + Var)$;
while *SAT[*S_i*]* **do**
  $ckt\_response = get\_circuit\_response(W_i)$;
  if $ckt\_response > oracle\_max$ **then**
    $S_{i+1} = S_i \wedge (W < W_i)$;
  else if $ckt\_response < oracle\_min$ **then**
    $S_{i+1} = S_i \wedge (W > W_i)$;
  else
    candidate_keys.append(W_i);
    $S_{i+1} = S_i \wedge (W\ != W_i)$;
end
return candidate_keys;

TABLE IV

RESULTS OF EXECUTING THE MA ON THE FIVE OBFUSCATED ANALOG BENCHMARK CIRCUITS DESCRIBED IN SECTION III AND FOR THE FOUR DIFFERENT SCENARIOS DESCRIBED IN SECTION IV

Scenarios	VGA (10-bits)			Mixer (10-bits)			OpAmp (12-bits)			2-Stage VGA (12-bits)			Front-End (18-bits)		
	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time
I	1%	1	0.22 s	1%	1	0.16 s	1%	2	0.33 s	1%	9	0.66 s	1%	313	12.58 s
	10%	3	0.26 s	10%	1	0.21 s	10%	4	0.36 s	10%	115	2.73 s	10%	2939	425.78 s
	25%	16	0.41 s	25%	30	0.63 s	25%	14	0.67s	25%	373	8.91 s	25%	9312	3051.84 s
II	1%	1	0.27 s	1%	1	0.15 s	1%	2	0.31 s	1%	1	0.50 s	1%	297	8.34 s
	10%	3	0.26 s	10%	1	0.16 s	10%	4	0.29 s	10%	38	1.13 s	10%	3097	181.98 s
	25%	16	0.46 s	25%	30	0.63 s	25%	15	0.61 s	25%	119	2.80 s	25%	8465	856.53 s
III	1%	1	0.23 s	1%	1	0.25 s	1%	2	0.35 s	1%	9	0.65 s	1%	417	12.55 s
	10%	3	0.24 s	10%	1	0.23 s	10%	4	0.42 s	10%	110	2.46 s	10%	1659	66.62 s
	25%	16	0.45 s	25%	36	0.86 s	25%	16	0.65 s	25%	372	9.46 s	25%	3538	198.64 s
IV	1%	0	65.08 s	1%	0	149.07 s	1%	0	62.66 s	1%	0	86.64 s	1%	0	119.53 s
	10%	276	3570.75 s	10%	0	148.58 s	10%	0	62.62 s	10%	4 No Correct Key	90.97 s	10%	7 No Correct Key	200.33 s
	25%	279	3662.48 s	25%	11 No Correct Key	308.42 s	25%	0	65.26 s	25%	10 No Correct Key	94.66 s	25%	13 No Correct Key	207.11 s
				90%	22	326.41 s	99%	0	62.26 s	90%	144 No Correct Key	2053.98 s	90%	10 No Correct Key	293.83 s

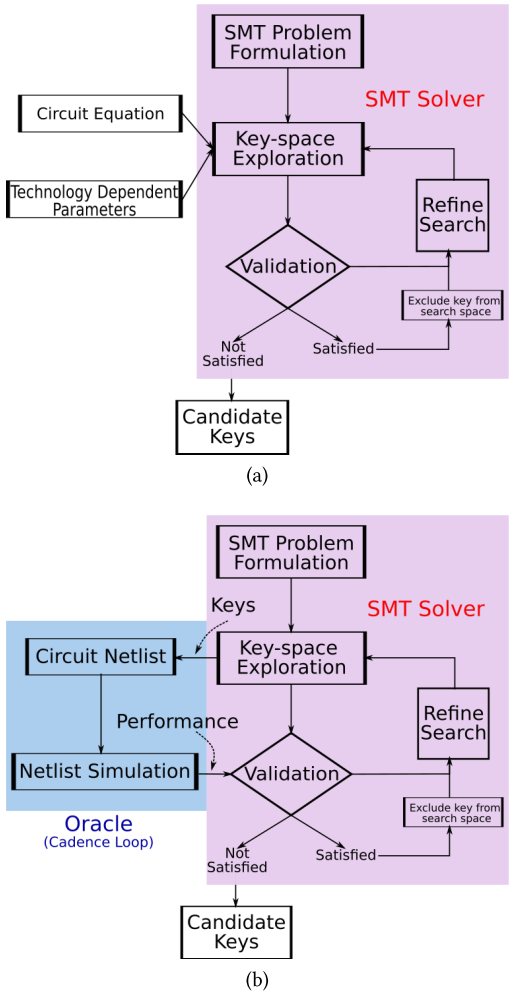

Fig. 6. Flowchart of the SMT-based MA executed for each of the four scenarios described in Section V. (a) Circuit equations are used to obtain input–output responses for Scenarios I–III. (b) Cadence-based oracle circuit emulation is used to obtain input–output responses for Scenario IV. Here, the magenta box highlights the steps performed by the SMT solver and the blue box highlights the steps performed in Cadence.

VII. METRICS OF EVALUATION

Four analog attack algorithms are evaluated on five analog circuit topologies for four attack scenarios representing a varying degree of circuit information possessed by the attacker. The four analog attack algorithms are evaluated utilizing both a performance metric T_P and a resources metric T_R. The performance metric evaluates the time required to successfully attack an analog circuit as well as the number of candidate keys returned that an attacker must then brute-force to determine the correct key. The resource metric provides an analysis of the time required to obtain the necessary information needed by the attacker to perform a successful attack on an analog circuit. Based on the performance and resource metrics, an overall attack evaluation metric T is formulated, which characterizes the efficacy and efficiency of the four analog attack algorithms.

A. Resource Metric (T_R)

The resource metric accounts for the time required to initialize the attack. The considered analog attack algorithms require prior knowledge of the obfuscated circuit and the implemented obfuscation methodology to effectively execute. The resource metric accounts for the total time required to collect all the information needed to perform the attack. The resources needed to effectively execute the four analog attack algorithms on the five considered analog circuits are listed in Table V. The cost and the time to obtain each resource are correlated to the number of steps each requires.

The number of steps needed to initialize an attack on a locked analog circuit is listed in Table VI. From the discussion provided in Section III and in [19], reverse engineering the circuit to obtain the netlist involves the most number of steps and, therefore, is considered to be the most expensive and time consuming. The time to obtain the circuit netlist is denoted as T_N. The delayering and probing methodology needed to obtain the biasing information, assuming that the attacker has already acquired the circuit netlist, involves the next highest number of steps, and is, therefore, the second most expensive and time consuming. The time to obtain the biasing information is denoted as T_B. The modeling of the obfuscated analog circuit using standard equations is the third most expensive and time consuming, provided

TABLE V

EVALUATION OF THE RESOURCE METRIC FOR THE FOUR ANALOG DEOBFUSCATION ATTACKS. A CHECK MARK IMPLIES NEEDED INFORMATION. THE NUMBER OF STEPS TO DETERMINE ALL NEEDED INFORMATION IS LISTED, WHERE EACH STEP REPRESENTS ONE UNIT OF TIME

Attack Techniques	Required Information					Resource Metrics (Unit-time)
	Netlist	Biasing Information	Models	PDK Information	Circuit Specifications	
SAT [11–13, 28]	✓	✗	✗	✗	✓	7
SMT [8]	✓	✓	✓	✓	✓	17
GA [10]	✓	✓	✗	✓	✓	14
Key Spacing [9]	✓	✗	✗	✗	✗	6
Monotonic Attack [9]	✗	✗	✓	✗	✓	4

TABLE VI

NUMBER OF STEPS NECESSARY FOR AN ADVERSARY TO DETERMINE NEEDED CIRCUIT INFORMATION TO EXECUTE AN ATTACK. THE NUMBER OF STEPS REQUIRED TO COMPLETE EXTRACTION OF THE BIASING VOLTAGES AND CURRENTS, CIRCUIT ATTACK MODEL(S), AND THE BIAS INFORMATION ASSUMES THAT THE ATTACKER POSSESSES THE CIRCUIT NETLIST

Information	Steps To Determine The Information	Number of Steps
Circuit Netlist	• Target IC • Depackaging • Delayering • Imaging • Extraction • Annotation	6
Biasing Voltages and Currents	• IC for Probing • Circuit Netlist • Decapsulation • Wire Location • Probing	5
Circuit Attack Model(s)	• Oracle IC • Input-Output Responses • Circuit Modelling	3
PDK/ Process Information	• Circuit Netlist • PDK Documentation	2
Circuit Specifications	• Datasheet	1

that the circuit netlist is already determined, and is denoted as T_M. The process of obtaining the PDK information is considered the fourth most expensive. Although negligible, the time to obtain the PDK information from the extracted circuit netlist and the PDK documentation is denoted as T_P. The circuit specifications are the simplest and cheapest information to obtain, which involves access to the datasheets. For the analysis of the circuit specifications, a single stage analog circuit is assumed obfuscated rather than a subblock of a larger circuit, where only system level specifications are provided. The time required to obtain circuit specifications is denoted as T_S.

The time needed to determine various information is subjective and is dependent on the process technology node, packaging type, number of metal layers, availability of PDK information, and the availability of circuit specifications. Therefore, an alternative method utilizing the number of steps required to gather each piece of needed information is proposed, with results as listed in Table VI. The total number of steps to determine all needed information for each deobfuscation attack is listed in Table V. Another characterization of the results listed in Table V is to assume that all steps required to gather the necessary information are treated equally and are, therefore, represented as one unit of time. The calculated resource metric for the SMT, GA, KS, and monotonic response attacks is 17, 14, 6, and 4 time units, respectively, as listed in Table V.

TABLE VII

TOTAL TIME REQUIRED TO OBTAIN ALL INFORMATION NEEDED TO DEOBFUSCATE AN ANALOG CIRCUIT WITH EACH ATTACK TECHNIQUE

Attack Technique	Time to Obtain Required Information
SMT	$T_{R_{SMT}} = T_N + T_B + T_M + T_P + T_S$
GA	$T_{R_{GA}} = T_N + T_B + T_P + T_S$
Key Spacing	$T_{R_{KS}} = T_N$
Monotonic Attack	$T_{R_{MA}} = T_M + T_S$

The proposed resource metric is evaluated to characterize the additional steps and resources required to perform the deobfuscation attack on a secured analog circuit. Unlike digital deobfuscation attacks that only require the netlist and input–output responses of the circuit, analog deobfuscation attacks require additional information, including the biasing conditions and accurate models of the circuit. The additional resources needed to execute an analog deobfuscation attack are listed in Table V while also evaluating the resource metric on a digital SAT attack for comparison.

As compared to digital SAT attacks, the SMT and GA attacks require 10 and 7 additional units of time, respectively, to successfully determine all the information needed to perform the given attack. The execution of the GA attack requires the reverse engineered obfuscated netlist, the bias information, and an oracle IC [10], which results in 14 total steps and, therefore, 14 units of time to initialize the attack. The SMT attack requires information on the circuit netlist, the bias voltages and/or currents, the PDK documentation, and the circuit specifications [8] to model the obfuscated circuit and formulate the SMT problem correctly. Obtaining all the necessary information to initialize the SMT attack requires a total of 17 steps and, therefore, 17 units of time, as indicated in Table V.

The KS attack and monotonic response attack require, respectively, 1 and 3 fewer time units as compared to the digital SAT attack to determine all information needed to execute the deobfuscation attack. The MA requires only the model that represents the monotonic output response of the circuit and the circuit specifications. Obtaining the two accounts for a total of four steps as indicated in Table V and is, therefore, the cheapest and the fastest algorithm to initialize. The KS attack is the second cheapest attack algorithm to initialize and execute. From analyzing the results listed in Tables V and VII, the resource metric T_R evaluated for each of the analog attack techniques results in a cost order of

$$T_{R_{\mathrm{MA}}} < T_{R_{\mathrm{KS}}} < T_{R_{\mathrm{SMT}}} < T_{R_{\mathrm{GA}}}. \quad (1)$$

TABLE VIII

EVALUATION OF THE PERFORMANCE METRIC FROM THE SMT, GA, KS, AND MONOTONIC RESPONSE ATTACKS EXECUTED ON A VGA, 2-STAGE VGA, AND FRONT-END. THE PERFORMANCE METRIC IS DETERMINED AS THE SUM OF THE EXECUTION TIME NEEDED TO DETERMINE THE CANDIDATE KEYS AND THE TOTAL TIME NEEDED TO PERFORM A BRUTE-FORCE ATTACK ON THE RETURNED CANDIDATE KEYS, WHICH IS GIVEN AS THE PRODUCT OF THE NUMBER OF CANDIDATE KEYS AND THE TIME REQUIRED TO PERFORM A TRANSIENT SIMULATION ON A SINGLE CANDIDATE KEY

Deobfuscation Attacks	Scenarios	VGA (10-bit key)			2-Stage VGA (12-bit key)			Front-End (18-bit key)		
		Ex. Time(s)	Candidate Keys	**Perf. Metrics**	Ex. Time(s)	Candidate Keys	**Perf. Metrics**	Ex. Time(s)	Candidate Keys	**Perf. Metrics**
SMT	I	6.30	1009	**3436.9**	43.60	3969	**25842.1**	2818.25	42948	**352844.45**
	II	6.74	1009	**3437.34**	51.07	3969	**25849.57**	48060	250047	**2085943.05**
	III	3.75	511	**1741.15**	92.63	3969	**25891.13**	52812	250047	**2090695.05**
GA	I	26640	1	**26643.4**	23652	-	-	432000	-	-
	II	26640	1	**26643.4**	23652	-	-	411480	-	-
	III	47880	1	**47883.4**	23652	-	-	411480	-	-
Key Spacing	I	1.93	19	**66.53**	0.89	27	**176.39**	1.25	36	**294.65**
	II	1.93	19	**66.53**	0.89	27	**176.39**	1.25	36	**294.65**
	III	1.93	19	**66.53**	0.89	27	**176.39**	1.25	36	**294.65**
Monotonic Response	I	0.41	16	**54.81**	8.91	373	**2433.41**	3051.84	9312	**78944.64**
	II	0.46	16	**54.86**	2.8	119	**776.3**	856.53	8465	**69846.28**
	III	0.45	16	**54.85**	9.46	372	**2427.46**	198.64	3538	**29033.34**

B. Performance Metric (T_P)

The performance metric accounts for the time to determine the candidate keys from the obfuscated analog circuit and the time to determine the correct key from the candidate keys when executing a brute-force attack. The time to determine the candidate keys is denoted as T_{Solve} and the time to determine the correct key from the candidate keys executing a brute-force attack is denoted as T_{BF}. The results provided in Section VI for the SMT, GA, KS, and monotonic response attacks are utilized to calculate the performance metric.

The results from the evaluation of the performance metric for the SMT, GA, KS, and monotonic response attacks and for Scenarios I–III are provided in Table VIII. Results are provided for a single stage VGA, a two stage VGA, and a three stage front-end circuit that are obfuscated with a 10-, 12-, and 18-bit key, respectively. The execution of the brute-force attack is performed through transient analysis of the analog circuit with each candidate key applied to the circuit using the Cadence schematic editor and Spectre simulation running on a server that includes two 12-core Intel Xeon E5 CPUs and 96 GB of DDR4 memory. The time required to perform each transient simulation on the server for each candidate key applied to a single stage, two stage, and multistage circuit is 3.4, 6.5, and 8.15 s, respectively. The time to input the candidate key on the obfuscated analog circuit is ignored. The performance metrics is computed as the time to execute a 10-µs transient simulation on the schematic of an obfuscated analog circuit using the Spectre simulator. Therefore, the performance metric is calculated as the sum of the attack execution time and the product of the number of candidate keys multiplied with the time required to perform each transient simulation on a candidate key.

The results listed in Table VIII indicate that for a single stage circuit obfuscated with a 10-bit key, the MA is the fastest and returns the least number of candidate keys and, therefore, provides the lowest performance score, where a lower score is desired. However, for a multistage circuit with keys exceeding 10 bits, the KS attack is the fastest and returns the least number of candidate keys. The SMT attack is the third fastest attack for determining the candidate keys for both single and multistage circuits. The GA attack returns a single candidate key, but the key is not guaranteed to be correct, and the attack either timed out after five days (432 000 s) or did not return a correct key for both the two stage VGA and the analog front-end for key sizes greater than 10 bits. In addition, the execution time of the GA attack is the longest, as the attack is dependent on results from Spectre simulations to optimize the CF. Therefore, (2) and (3) provide a summary of the results from the characterization of the performance metric when executing the attack algorithms on single stage and multistage obfuscated circuits, respectively

$$T_{P_{\text{MA}}} < T_{P_{\text{KS}}} < T_{P_{\text{SMT}}} < T_{P_{\text{GA}}} \tag{2}$$

$$T_{P_{\text{KS}}} < T_{P_{\text{MA}}} < T_{P_{\text{SMT}}} < T_{P_{\text{GA}}}. \tag{3}$$

C. Attack Evaluation Metric (T)

The attack evaluation metric is defined as the summation of the resource metric with the performance metric. The attack evaluation metric is utilized to rank the analog attack techniques for two criteria. The first criterion accounts for Scenario I and considers the conditions described in [8], [9], and [10] for the SMT attack, GA attack, and both the KS attack and MA, respectively. For the first criterion, the time to obtain the circuit netlist far outweighs both the time to obtain any other information and the time to determine the correct key. Therefore, for Scenario I, the assessment of the attack evaluation metric on the different attack techniques results in a cost order of

$$T_{\text{MA}} < T_{\text{KS}} < T_{\text{SMT}} < T_{\text{GA}}. \tag{4}$$

For the second criterion, only the circuit netlist and the circuit specifications are assumed available. For the proposed scenario, the assessment of the attack evaluation metric results in a cost order of

$$T_{\text{KS}} < T_{\text{MA}} < T_{\text{SMT}} < T_{\text{GA}}. \tag{5}$$

The results from the analysis of the attack evaluation metric suggest that the execution of the MA is the fastest and the

cheapest to deobfuscate a locked analog circuit. The MA is considered a black-box attack, as only the oracle circuit and high-level circuit specifications are required. However, if the locked analog circuit consists of multiple subblocks that produce a nonmonotonic parameter response, the MA fails to determine the correct key.

When the MA fails, time and resources are invested to extract the locked netlist of the analog circuit through reverse engineering. Once the circuit netlist is obtained, the execution of the KS attack is the fastest and cheapest to deobfuscate the locked analog circuit. The KS attack only requires the circuit netlist. However, the KS attack is only applicable to an analog circuit obfuscated through key-based obfuscation. For analog circuits secured with nonkey-based obfuscation including multithreshold obfuscation, attacks based on the SMT and GA techniques are applicable.

Although the GA attack executes for Scenario I, a comparison of the calculated performance metric for each attack indicates that the GA attack ($T_{P_{\text{GA}}}$) is costlier than the SMT attack ($T_{P_{\text{SMT}}}$) as the SMT attack requires less resources and is faster to execute than the GA attack. In addition, the GA attack failed to return the correct key for multistage locked analog circuits, as indicated by the results listed in Table II. Furthermore, the execution time of the GA attack is approximately 50 to 300 times longer than that of the SMT attack for single stage circuits when considering Scenario I. The execution time of the GA is further reduced by using less complex circuit simulators such as PySpice at the expense of accuracy. To compensate for less accurate circuit simulators, larger population and generation sizes are required that increase the number of iterations needed to complete the GA attack, which results in a higher number of iterations of the GA attack than that of the SMT attack.

VIII. Discussion

The analysis of evaluating the four analog attack techniques on five distinct analog circuits indicates that the SMT attack performs exceptionally well in determining the correct key under Scenario I, where the attacker possesses all the circuit information except for the correct key. However, when one or more additional pieces of information is missing, the attack is costly and ineffective, where the SMT attack pruned only 1.5% of the 10-bit keys obfuscating a single stage mixer characterized for Scenarios II and III. In addition, as compared to the monotonic and the KS attacks, the SMT attack is slower by 224× and 2240×, respectively.

The GA attack determined the correct key for the single stage circuits in less than 8 h but failed to determine the correct key for multistage circuits even when executing an equal number of iterations as a brute-force attack. The KS attack is efficient in pruning keys, as the attack is 10× faster than the MA and results in 8.3× fewer candidate keys for multistage analog circuits but requires the circuit netlist. The MA determined the correct key in less than 1 s for single stage circuits, less than 10 s for two stage circuits, and less than 1 h for three stage circuits. As the number of stages increases and for Scenario IV, the circuit response is nonmonotonic, which results in the failure of both the GA and monotonic response attacks in determining the correct key.

Metrics to evaluate the performance of the four analog attack techniques are developed based on the attack initialization time and the attack execution time. The evaluation of the metrics indicates that the MA is the fastest and the cheapest attack to execute when the attacker possesses no circuit information except the circuit specifications, but fails for multistage circuits with nonmonotonic responses. However, when the attacker possesses the circuit netlist, the KS attack is the fastest and the cheapest attack to deobfuscate a multistage obfuscated analog circuit. Execution of the SMT attack is faster than the KS attack for a single stage analog circuit attacked under Scenario I, but the execution time and the number of candidate solutions increase exponentially for an obfuscated multistage analog circuit. The execution time of the GA is the longest, where the GA attack requires 1 21 091× longer time to complete than the MA even for a single stage analog circuit attacked under Scenario I.

IX. Conclusion

In this article, the practical performance of four state-of-the-art analog attack algorithms is evaluated for four threat scenarios. The challenges of determining the necessary information needed to efficiently execute an attack on an obfuscated circuit are described. Four different threat models are developed and analyzed, where an adversary is assumed to possess progressively less information pertaining to the analog circuit and the implemented security features. The performance of the analog attack algorithms is evaluated on five distinct analog circuits obfuscated with a key-based parameter locking technique and with varying key sizes and a varying number of locked stages. In addition, novel metrics based on the resources needed to initialize the attack and the time to execute the attack algorithms are proposed. The evaluation of the metrics indicates that the MA is the fastest to deobfuscate a circuit, requiring less than 10 h to determine the correct key of single stage circuits. The KS attack is the fastest and cheapest to execute when deobfuscating multistage analog circuits, but requires the circuit netlist. The SMT attack is faster than the KS attack for a single stage analog circuit attacked under Scenario I, but the execution time and the number of candidate keys returned increase exponentially for an obfuscated multistage analog circuit. The execution time of the GA is the longest, where the attack timed out after five days without determining the correct key of a locked two stage VGA and front-end circuit even for Scenario I. With the performance analysis of the four attack algorithms and the development of metrics based on the attack algorithms, analog circuit designers are provided with insight on different threat models to consider when securing analog circuits, and the research community is encouraged to consider these varying threat models when developing both novel techniques to protect and attack analog circuits.

References

[1] Y. Bi, J. Yuan, and Y. Jin, "Beyond the interconnections: Split manufacturing in RF designs," *Electronics*, vol. 4, no. 3, pp. 541–564, Aug. 2015.

[2] V. V. Rao and I. Savidis, "Performance and security analysis of parameter-obfuscated analog circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 12, pp. 2013–2026, Dec. 2021.

[3] D. H. K. Hoe, J. Rajendran, and R. Karri, "Towards secure analog designs: A secure sense amplifier using memristors," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2014, pp. 516–521.

[4] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting analog IC piracy via combinational locking," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2017, pp. 1–10.

[5] 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 Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2018, pp. 1–8.

[6] J. Leonhard et al., "MixLock: Securing mixed-signal circuits via logic locking," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 84–89.

[7] A. A. Saki and S. Ghosh, "How multi-threshold designs can protect analog IPs," in *Proc. IEEE 36th Int. Conf. Comput. Design (ICCD)*, Oct. 2018, pp. 464–471.

[8] 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 Int. Test Conf. (ITC)*, Nov. 2019, pp. 1–10.

[9] V. V. Rao, K. Juretus, and I. Savidis, "Security vulnerabilities of obfuscated analog circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[10] R. Y. Acharya, S. Chowdhury, F. Ganji, and D. Forte, "Attack of the genes: Finding keys and parameters of locked analog ICs using genetic algorithm," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Dec. 2020, pp. 284–294.

[11] K. Juretus and I. Savidis, "Reduced overhead gate level logic encryption," in *Proc. Int. Great Lakes Symp. VLSI (GLSVLSI)*, May 2016, pp. 15–20.

[12] M. Yasin and O. Sinanoglu, "Evolution of logic locking," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Oct. 2017, pp. 1–6.

[13] M. Yasin, J. J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 9, pp. 1411–1424, Sep. 2016.

[14] M. Elshamy, "Design for security in mixed analog-digital integrated circuits," Ph.D. thesis, Jul. 2021.

[15] S. Y. Lim, S. P. Phang, T. Trupke, A. Cuevas, and D. Macdonald, "Dopant concentration imaging in crystalline silicon wafers by band-to-band photoluminescence," *J. Appl. Phys.*, vol. 110, no. 11, pp. 1–8, Dec. 2011.

[16] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *Proc. 18th IEEE Latin Amer. Test Symp. (LATS)*, Mar. 2017, pp. 1–6.

[17] V. V. Rao and I. Savidis, "Mesh based obfuscation of analog circuit properties," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[18] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1601–1618.

[19] S. E. Quadir et al., "A survey on chip to system reverse engineering," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 1, pp. 1–34, Jan. 2017.

[20] F. Courbon, "Practical partial hardware reverse engineering analysis," *J. Hardw. Syst. Secur.*, vol. 4, no. 1, pp. 1–10, Mar. 2020.

[21] A. F. J. Levi and G. Aeppli, "The naked chip: No trade secret or hardware trojan can hide from ptychographic X-ray laminography," *IEEE Spectr.*, vol. 59, no. 5, pp. 38–43, May 2022.

[22] J. Kumagai, "Chip detectives [reverse engineering]," *IEEE Spectr.*, vol. 37, no. 11, pp. 43–48, Nov. 2000.

[23] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: Challenges and research opportunities," *IEEE Des. Test*, vol. 34, no. 5, pp. 63–71, Oct. 2017.

[24] L. Lewyn and N. Williams, "Is a new paradigm for nanoscale analog CMOS design needed?" *Proc. IEEE*, vol. 99, no. 1, pp. 3–6, Jan. 2011.

[25] X. Zhu, Y. Yang, and H. Liu, "Cell-based wideband variable-gain amplifier with accurate gain adjustment in 65 nm CMOS technology," in *Proc. IEEE Int. Workshop Electromagn., Appl. Student Innov. Competition (iWEM)*, May 2016, pp. 1–3.

[26] B. Razavi, "Operational amplifier," in *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2017, ch. 9, pp. 344–409.

[27] B. Razavi, "Phase-locked loops," in *Design of Analog CMOS Integrated Circuits*, New York, NY, USA: McGraw-Hill, 2017, ch. 16, pp. 651–690.

[28] Y. Zhong and U. Guin, "Complexity analysis of the SAT attack on logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, pp. 1–14, Jan. 2023, doi: 10.1109/TCAD.2023.3240933.

Vaibhav Venugopal Rao (Student Member, IEEE) received the B.E degree in electronics and communications from Visvesvaraya Technological University, Belagavi, India, in 2015, and the M.Sc. degree in computer engineering from Drexel University, Philadelphia, PA, USA, in 2017, where he is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering.

His current research interests include circuit-level techniques to protect analog IP and variation-aware automation of analog circuit design to reduce design time.

Kyle Juretus (Member, IEEE) received the Ph.D. degree in electrical engineering from Drexel University, Philadelphia, PA, USA, in 2020.

During his Ph.D. studies, he was a National Defense Science and Engineering (NDSEG) Fellow sponsored by the Air Force Research Laboratory (AFRL) from 2016 to 2019. He is currently an Assistant Professor with the Electrical and Computer Engineering Department, Villanova University, Villanova, PA, USA. His research focuses on integrated circuit (IC) design with the objective to create faster, smaller, more efficient, and more secure ICs. To enable such changes with minimal design time overhead, he focuses on the development of next-generation electronic design automation (EDA) tools.

Ioannis Savidis (Senior Member, IEEE) received the B.S.E. degree in electrical and computer engineering and biomedical engineering from Duke University, Durham, NC, USA, in 2005, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, USA, in 2007 and 2013, respectively.

He joined the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA, in 2013, where he is currently an Associate Professor and directs the Integrated Circuits and Electronics (ICE) Design and Analysis Laboratory. His current research interests include analysis, modeling, and design methodologies for high-performance digital and mixed-signal integrated circuits, power management for system-on-a-chip (SoC) and microprocessor circuits, hardware security, including digital and analog obfuscation and Trojan detection, and electrical and thermal modeling and characterization, signal and power integrity, and power and clock delivery for heterogeneous 2-D and 3-D circuits.

Dr. Savidis is a member of the Association of Computing Machinery, the IEEE Circuits and Systems Society, the IEEE Communications Society, and the IEEE Electron Devices Society. He was a recipient of the 2018 National Science Foundation Early Faculty (CAREER) Award. He serves on the Organizing Committee of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST), the ACM Great Lakes Symposium on VLSI (GLSVLSI), and the IEEE International Symposium on Circuits and Systems (ISCAS). He serves on the Editorial Board of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the *Microelectronics* journal, and the *Journal of Circuits, Systems and Computers*.