Breaking Analog Locking Techniques

Nithyashankari Gummidipoondi Jayasankaran, *Student Member, IEEE*, Adriana Sanabria-Borbón, *Member, IEEE*, Amr Abuellil, Edgar Sánchez-Sinencio, *Life Fellow, IEEE*, Jiang Hu, *Fellow, IEEE*, and Jeyavijayan Rajendran, *Senior Member, IEEE*

***Abstract*— Similar to digital circuits, analog circuits are also susceptible to supply-chain attacks. There are several analog locking techniques proposed to combat these supply-chain attacks. However, there exists no elaborate evaluation procedure to estimate the resilience offered by these techniques. Evaluating analog defenses requires the usage of non-Boolean variables, such as bias current and gain. Hence, in this work, we evaluate the resilience of the analog-only locks and analog and mixed-signal (AMS) locks using satisfiability modulo theories (SMTs). We demonstrate our attack on five analog locking techniques and three AMS locking techniques. The attack is demonstrated on commonly used circuits, such as bandpass filter (BPF), low-noise amplifier (LNA), and low-dropout (LDO) voltage regulator. Attack results on analog-only locks show that the attacker, knowing the required bias current or voltage range, can determine the key. Likewise, knowing the protected input patterns (PIPs), the attacker can determine the key to unlock the AMS locks. We then extend our attack to break the existing analog camouflaging technique.**

***Index Terms*— Analog locking, hardware security, IP protection, logic locking.**

I. Introduction

AS BUILDING an integrated circuit (IC) fabrication unit costs billions of dollars, most companies have gone fabless [1]. Fabrication outsourcing has led to the vulnerability of supply-chain attacks, such as intellectual property (IP) piracy, counterfeiting, overproduction, reverse engineering (RE), and hardware Trojan insertions [1]. The works [2] and [3] propose several design-for-trust (DfTr) techniques such as IC metering, watermarking, logic locking, split manufacturing, and camouflaging. These techniques help to thwart the supply-chain attacks. Existing DfTr techniques mostly target digital ICs. However, analog ICs are more prone to supply-chain attacks than digital ICs as they are easier to reverse engineer [4]. This high vulnerability is due to their low transistor count compared to their digital counterparts. They also have predefined layout patterns, e.g., common-centroid, to tolerate process variations [5].

Manuscript received January 14, 2020; revised May 30, 2020; accepted June 14, 2020. This work was supported by the National Science Foundation under Grant CCF-1815583. *(Corresponding author: Nithyashankari Gummidipoondi Jayasankaran.)*

Nithyashankari Gummidipoondi Jayasankaran, Adriana Sanabria-Borbón, Amr Abuellil, Edgar Sánchez-Sinencio, and Jeyavijayan Rajendran are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: gjn@tamu.edu; adca.sanabria@tamu.edu; aabuellil@tamu.edu; s-sanchez@tamu.edu; jv.rajendran@tamu.edu@tamu.edu).

Jiang Hu is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA, and also with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: jianghu@tamu.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2020.3007159

A. Related Works on Analog Locking

Different DfTr techniques based on logic locking and camouflaging have been developed for analog ICs [6]–[10]. We shall now discuss the existing analog-only locking schemes. Researchers use a memristor-based voltage divider, which tunes the body-bias voltage required for offset voltage cancellation in sense amplifiers [6]. The memristor's configurations are given only to the authorized user. Another work proposes a satisfiability modulo theory (SMT)-based combinational lock [7]. A configurable current mirror (CCM), whose output current is controlled by the key inputs, produces the bias current (I_B). Circuit specifications, such as center frequency (f_c) and oscillation frequency (ω_{osc}), depend on the precise value of I_B. Similar to [7], the effective width of the transistor depends on the key inputs in parameter-biasing obfuscation [8], which determines the bias.

A trained analog neural network (ANN) provides the necessary bias voltages to the low-noise amplifier (LNA) [10]. Only a unique set of input voltages can give the desired bias value, as the ANN is preprogrammed with fixed weights. In [9], analog circuits are camouflaged by replacing the nominal threshold voltage (V_{th}) transistors (NVT) with resized high-V_{th} (HVT) and low-V_{th} (LVT) transistors. The NVT transistors control the circuit specifications, such as f_c, bandwidth (BW), and ω_{osc}. The high-resolution pictures of the depackaged chip cannot reveal the V_{th} type of transistors; thus, the original design cannot be recovered.

B. Related Works on Analog and Mixed-Signal Locking

Techniques for protecting analog and mixed-signal (AMS) circuits have been proposed in [11]–[14]. The work in [11] consists of an analog circuit and a logic-locked optimizer. The key input controls the working of the optimizer. This optimizer sets the correct value of the passive components in the analog circuit, which enables the desired circuit performance. Similarly, in MIXlock, unless a correct key is given, the locked digital circuit sets one or more circuit specifications of the analog circuit outside the acceptable range [12]. This technique is demonstrated on a $\Sigma\Delta$ analog-to-digital converter (ADC) [13]. In shared dependencies, the analog and the digital parts of the AMS circuits are locked using parameter-biasing obfuscation [8] and stripped functionality logic locking (SFLL)-HD^0 [3], respectively [14].

C. Attacks Against Digital Logic Locking

SAT attack is a Boolean satisfiability-based attack on combinational logic-locked circuits [15]. It uses the SAT solver to weed out incorrect keys. SMT attack [16] that is a superset of SAT attack, can handle non-Boolean variables, e.g., logic delay. This attack can break the delay logic locking [17]. Removal attack identifies the protection logic and removes

it to recover the original circuit [18]. Bypass attack finds the distinguishing input patterns (DIPs) that give an incorrect output for an incorrect key. The attacker adds a bypass circuitry around the protection block to restore the output for those DIPs [19]. The attacks such as SFLL-hd–Unlocked [20] and FALL attack [21] break SFLL-HD^h [3]. These attacks determine the protected input patterns (PIPs) by structurally analyzing the locked netlist. It then extracts the key using the determined PIPs.

D. Limitations of Existing Attacks

The attacks on digital locks cannot be applied on analog-only locks because the following holds.

1) The output of the analog-only locks is a non-Boolean variable, such as bias current and bias voltage. As the abovementioned attacks can handle only Boolean variables, they cannot break analog-only locks.
2) The bias circuit is locked using the techniques proposed in [6]–[8]. Launching removal attacks removes the bias required for the circuit to be functional.
3) The bypass attack replaces the locked bias circuit with the precise current/voltage source. It is feasible only if the attacker knows the precise bias inputs and can pirate the design. Otherwise, it is infeasible.
4) The SMT formulation in [16] targets the delay logic locking [17] and speeds up the SAT attack. The constraints to model the analog locks are different from delay logic locking. Hence, new SMT formulations are required to break the analog-only locks.

Also, the SFLL-HD^h used in [11] and [12] is resilient against all the attacks mentioned above, except [20] and [21]. In [20] and [21], for a given input size, key size, and Hamming distance (HD), all possible PIPs are considered to determine the key. However, as detailed in Section II, the attacker does not have access to all possible PIPs. Hence, we need updated SAT formulations that can return the correct key. Therefore, there arises a need for developing an evaluation technique for existing analog-only and AMS locks. In this article, we focus on building such evaluation techniques. New SMT (SAT) formulations are developed to break the analog [6]–[10] (AMS [11]–[14]) locks.

E. Naïve Manual Attack Against Analog Locking

Attacking a lock of analog IC like [7] means to figure out the correct key or configuration of the locking circuit. Conceivably, there are two naïve approaches: 1) brute-forcing all key combinations and 2) redesigning the circuit without the lock. The former approach takes exponential time with respect to key size, which is prohibitively expensive. The latter entails high design expertise and design effort. As such, none of them is appealing to attackers pursuing fast and cheap solutions.

F. Our Approach and Contributions

In this work, we propose the SMT- and the SAT-based attacks to break the analog and digital locks in AMS circuits, respectively. Based on the information collected from various sources tabulated in Table I, an attacker can find the correct key for proper circuit operation using SMT and SAT formulations. We have developed attacks to break: 1) analog-only locks [6]–[10]; 2) digital locks in AMS circuits [11], [13], [14]; and 3) analog locks in AMS circuits [14]. The contributions of this article are as follows.

1) We propose new SMT formulations to break analog locks [6]–[10].

TABLE I
SOURCES OF INFORMATION AVAILABLE TO THE ATTACKER

Source	Information acquired
Layout file from the foundry or the reverse engineered netlist using the oracle [22]	Sizes of passive components (R, C)
	Key size and transistor count
	W and L of the transistors
	Key connectivity to transistor switches or memristors
Technology library [23] (PDK documentation)	Values of passive components (R, C) and transistor details (μ, C_{ox}, t_{ox}, V_{th})
	Availability of different V_{th} transistors
Circuit specification [24]	Minimum and maximum values of I_B and V_B, which are output of the bias circuit
	Values of I_{ref} and V_{ref}, which are the input to the bias circuit
	Minimum and maximum values of the resistance that can be programmed into the memristors
	Values of circuit parameters (BW, ω_{osc})

2) We demonstrate our attack on the combinational locks [7] and the parameter-biasing obfuscation [8]. We demonstrate it on the following analog circuits due to their ubiquitous presence in wireless communication networks: Gm-C bandpass filter (BPF), LC oscillator, quadrature oscillator, and class-D amplifier.
3) We validate our attack on memristor-based protection [6].
4) We propose SAT formulations to break digital locks in AMS circuits and demonstrate this attack on [11].
5) We demonstrate the attack on the defense in [14], which requires both SMT and SAT formulations.
6) We extend our attack to evaluate analog camouflaging [9].

II. ATTACK APPROACH

A. Threat Model

As stated in Table I, we consider the following threat model, where both the foundry and the end-user are untrusted entities [2], [3], [11], [12], [15], [18]. The attacker in the untrusted foundry has access to the layout of the design provided by the designer, the process design kit (PDK) documentation, and the locking algorithm used as this information is public. He/she can overproduce the chip and sell the excess chips in the black market. Likewise, an end-user as an attacker has access to the RE tools to obtain the netlist of a locked chip [25]. It is relatively easy to reverse engineer analog circuits with several hundreds of transistors compared to SoCs with multimillion transistors. Also, compared to digital circuits, the analog circuits have a bigger transistor size [26] and predefined layout patterns, rendering them easier to reverse engineer. Similar to the attacker in the foundry, the untrusted end-user has access to the locking algorithm used. He/she also purchases a chip that has the correct key loaded. This chip serves as an oracle, where the attacker can observe the output for a given input. The manufacturer provides the specification along with the purchased chip; thus, an untrusted end-user can have access to it.

B. Attack Methodology on Analog Locks

The analog locks obfuscate the effective value of the circuit components, such as the width of the transistor, resistance,

and capacitance. These components are used in the bias circuit, as the precise bias current (I_B) or voltage (V_B) is required for the proper operation of the analog circuits. These components are called obfuscated components as they are made configurable, and their effective value is hidden from the attacker. The key input determines the effective values of these components. Only the correct key can set the effective values of the obfuscated components correctly; this is essential for precise biasing conditions and hence, the proper operation of the analog circuits. Our attack aims to find the key that gives the required bias to make the circuit functional. The attack methodology is described in the following.

1) *Identifying the Obfuscated Circuit Components and Their Dependence on the Key Inputs:* From the locked netlist, the attacker can determine the obfuscated components, such as transistors, resistors (Rs), and capacitors (Cs) [22], by tracing the wire connections from the key input. A component y in the original design is replaced by a set of n obfuscation components $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, which are controlled by an m-bit key vector $\mathbf{q} = (q_1, q_2, \ldots, q_m)$. We denote the values of y and x_i, $i \in \{1, 2, \ldots, n\}$, by y_v and $x_{i_v}, i \in \{1, 2, \ldots, n\}$, respectively. Then, the effective value of the obfuscated components $\mathbf{x}$ is
$$\tilde{y_v} = \phi(\mathbf{x_v}, \mathbf{q}). \tag{1}$$
Here, $\mathbf{x_v} = (x_{1_v}, x_{2_v}, \ldots, x_{n_v})$, and the function ϕ depends on how the obfuscation circuit is constructed. For the correct key $\mathbf{q}^*$, $\tilde{y_v} = y_v$, i.e., the effective value of the obfuscated component is equal to that of the original one.
2) *Finding the Equation Linking the Value of the Obfuscated Component $\tilde{\mathbf{y}}_\mathbf{v}$ to the Bias z_{ob_comp}:* The bias $z_{\text{ob\_comp}}$ (e.g., I_B or V_B) is a function ψ of the value of the obfuscated component $\tilde{y_v}$
$$z_{\text{ob\_comp}} = \psi(\tilde{y_v}). \tag{2}$$
Substituting (1) in (2) gives the dependence of $z_{\text{ob\_comp}}$ on the key $\mathbf{q}$, $z_{\text{ob\_comp}} = \psi(\phi(\mathbf{x_v}, \mathbf{q}))$.
3) *Derive Bias z_{spec} From the Circuit Parameter $\mathbf{p}$ of the Protected Analog IC:* The attacker can obtain the circuit parameters, such as g_m, BW, and ω_{osc}, from the circuit specification. He/she then analyzes the target circuit and extracts its characteristic equations. Solving these equations yields the bias point I_B or V_B
$$z_{\text{spec}} = \theta(p). \tag{3}$$
Here, θ is the function to compute the circuit parameter p [27]–[29]. If the attacker knows the bias point, he/she can redesign the entire analog circuit. However, it is sometimes possible to determine only the bias range and not the precise z_{spec}. This is because there may not be a direct equation linking z_{spec} and p. Instead, equations linking the minimum and maximum values of the bias with different circuit parameters are available in the specification [30]. Here, we calculate a range for bias using the equations $z_{\text{spec}_{\text{min}}} = \theta_1(p_1)$ and $z_{\text{spec}_{\text{max}}} = \theta_2(p_2)$
$$z_{\text{spec}_{\text{min}}} \leq z_{\text{spec}} \leq z_{\text{spec}_{\text{max}}} \tag{4}$$
where θ_1 and θ_2 are the functions to compute p_1 and p_2, respectively. $z_{\text{spec}_{\text{min}}}$ and $z_{\text{spec}_{\text{max}}}$ are the minimum and maximum values of the bias, respectively.

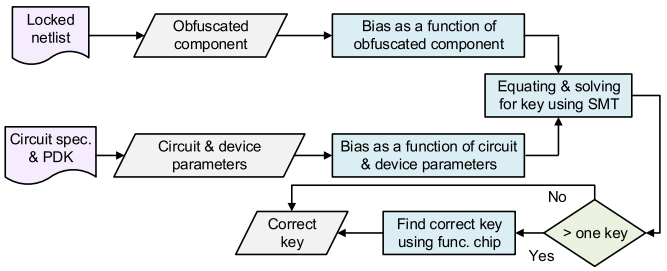

Fig. 1. Proposed SMT-based attack methodology. Circuit specification (Circuit spec.), PDK, and functional chip (func. chip).

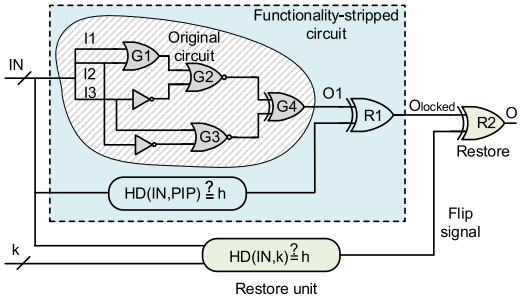

Fig. 2. Stripped-functionality logic locking [3]. HD, key (k), IN, PIP.

4) *Putting It All Together:* The I_B or V_B obtained from circuit specification and the I_B or V_B estimated from the obfuscated components should be equal or approximately equal for the analog circuit to be functional. Hence, the bias $z_{\text{ob\_comp}}$ equals z_{spec}, or alternatively, $z_{\text{ob\_comp}}$ satisfies the inequality specified by z_{spec}. Solving (1), (2), and (4) using the SMT-solver [31] computes the correct key $\mathbf{q}^*$. If the solver returns more than one key, the attacker compares the output response of an unlocked netlist for each of these keys with the oracle's response. He/she can choose the key that gives the desired or close to the desired output response. The correct key sets the effective value of the obfuscated component equal to the value of the original component, i.e., $\tilde{y_v} = y_v$.

The overall attack methodology is shown in Fig. 1.

C. Attack Methodology on Digital Logic Locking

SFLL [3] has been extensively used in locking the digital section of the AMS circuits in [11]–[14]. We will explain the working followed by the attack methodology on this technique.

1) Working: There are different variants in SFLL [3], such as SFLL-HD^0, SFLL-HD^h, SFLL-flex, and SFLL-fault. It protects only a certain number of input patterns (INs) called the PIPs. The output of the original circuit, O1, is inverted only when the IN is a PIP. The functionality-stripped circuit comprises of the original circuit, the inversion logic, and the logic which checks if the IN is a PIP. Depending on the variant of SFLL, the corruption injected by the inversion logic is restored, when the following holds.

1) The HD between the external key (k) and the IN equals 0 in SFLL-HD^0.
2) The HD between k and the IN equals h in SFLL-HD^h, as indicated in Fig. 2.
3) The input equals one of the PIPs that are stored in a content addressable memory in SFLL-flex.

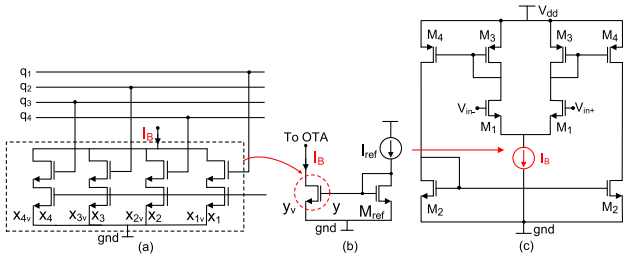

Fig. 3. CCMs to thwart IP piracy of analog circuits [7]. (a) Matrix of transistor switches which will replace the transistor y. (b) Original current mirror. (c) OTA with CCM supplying the bias current.

2) Attack Methodology: As SFLL protects only a handful of INs, the probability of finding them is negligible [3]. However, when this technique is used to lock the digital section of the AMS circuits, the PIPs can be found by analyzing the analog–digital interface signals. The key is then determined using SAT formulations, which are discussed next.

1) *Finding PIPs:* In [11]–[14], the analog section of the AMS circuit drives the input to the locked digital section. By simulating the analog section of the netlist, the attacker can determine the INs that drives the digital section. The inputs from the analog circuit are the only INs to the locked digital circuit. Hence, the PIPs should be the subset of these INs.
2) *SAT Formulation to Determine the Correct Key:* The attack formulations are unique based on the locking technique used [11], [14]. They are explained in the respective attack Sections II-H and II-I. The correct key is found by solving these formulations.

We first demonstrate our attack on analog locks such as combinational locks [7] and parameter-biasing obfuscation [8]. We then show how this attack methodology breaks other analog IP protection techniques [6], [8]–[10], [14]. Following this, the attack on digital logic locking techniques protecting the AMS circuits [11]–[14] is demonstrated.

D. Attack on Combinational Locks [7] and Parameter-Biasing Obfuscation [8]

Working: The defense technique in [7] proposes an SMT-based combinational locking technique. Circuit parameters, such as f_c, BW, and ω_{osc}, depend on the precise value of the bias current (I_B). A CCM generates this I_B. The key input configures the effective width of the mirroring transistor in the CCM. The transistor sizes are modeled using the SMT formulations such that on a correct key, the CCM gives the desired I_B. Otherwise, I_B is outside the range $((1-\Delta)I_B, (1+\theta)I_B)$, where Δ and θ are lower and upper bounds, respectively. The defender sets these bounds. A similar technique [8] randomly chooses the transistor sizes and hence can have more than one correct key. We consider the operational transconductance amplifier (OTA) shown in Fig. 3(c) as an example to demonstrate our attack. The current mirror supplies the required I_B to the OTA, as illustrated in Fig. 3(b). The transistor switch matrix replaces the mirroring transistor y, as shown in Fig. 3(a) and (b). The key-bits are connected to the gate terminal of the switches, controlling the magnitude of I_B. The SMT formulations required for this attack are:

1) *Obfuscated Component:* y_v is the ratio of (W/L) of transistor y with respect to (W/L) of M_{ref} in Fig. 3, where (W/L) is the aspect ratio of the transistor. In the CCM, the key input controls this effective size ratio ($\tilde{y_v}$). y is replaced by n (=4) NMOS transistor switches. The gate terminals of these switches are controlled by the key $\mathbf{q}$. x_{i_v} is the ratio of the aspect ratio of transistor x_i with respect to the aspect ratio of M_{ref}, where $i \in \{1,2,3,4\}$. The attacker can get the values x_{i_v} from the locked netlist, as shown in Table I. He/she also determines the key-bits that control each of the transistor switches, from the locked netlist. Hence, the obfuscated size ratio is $\tilde{y_v} = \sum_{i=1}^{n} x_{i_v} q_k$, where $q_k \in \{0,1\}$ and $k \in \{1,2,3,4\}$.
2) *Equations Linking the Bias z_{ob_comp} With Obfuscated Widths $\tilde{y1_v}$ and $\tilde{y2_v}$ of the Transistors:* In the analog locking techniques [6]–[8], the bias circuit is designed to be either a current mirror or voltage divider. If the bias circuit is a current mirror as in [7], I_B of CCM is $\psi(\tilde{y_v}) = I_B = \tilde{y_v} \times I_{\text{ref}}$, where I_{ref} is the reference current obtained from the circuit specification. For a voltage divider built using resistors y_1 and y_2, the bias voltage is determined by, $\psi(\tilde{y1_v}, \tilde{y2_v}) = V_B = (V_{\text{ref}} \times \tilde{y2_v})/(\tilde{y1_v} + \tilde{y2_v})$. Here, y_1 and y_2 are the original resistors replaced by a set of obfuscated resistors whose effective values are $\tilde{y1_v}$ and $\tilde{y2_v}$, respectively. Hoe *et al.* [6] realize y_1 and y_2 as resistors. Rao and Savidis [8] realize them as the resistance offered by transistor switches whose resistivity $y_v = (1/g_m) = (1/(2\mu C_{\text{ox}}(W/L)I_D)^{1/2})$, where the drain current $I_D = I_B/2$. g_m is the transconductance of the transistor, μ is the mobility of the transistor, and C_{ox} is the oxide capacitance. These values are available in the PDK [23]. V_{ref} is the reference voltage, which is obtained from the circuit specification [30].
3) *Equation Linking the Bias z_{spec} With Circuit Parameter p:* The g_m of the OTA shown in Fig. 3(c) is $(g_{m1}g_{m4}/g_{m3})$. Here, g_{mi} is the transconductance of transistor Mi. If x_v is the ratio of the aspect ratio of transistor $M4$ to the aspect ratio of $M3$, then $g_m = x_v \times g_{m1}$. The attacker finds I_B using $\theta(g_m) = I_B = (g_m^2/\mu C_{\text{ox}}(W/L))$. To calculate this desired I_B, he/she obtains the value of g_m from the specification, device parameters (μ and C_{ox}) from the PDK, and the transistor dimensions (W and L), from the netlist.
4) *Putting It All Together:* Solving the equations (A), (B), and (C) in Table II, gives the correct key $\mathbf{q}^*$. This key sets the required g_m in the OTA.

The abovementioned attack methodology can break the combinational locking technique [7]. The same methodology can also break the parameter-biasing obfuscation technique [8]. This is because the technique used in parameter-biasing obfuscation [8] is similar to [7]. In [8], the width of the transistor in the bias circuit is obfuscated. This is achieved by replacing the transistor with multiple transistors connected in parallel, as illustrated in Fig.3 (a) and (b). Hence, the attack methodology proposed for [7] can break [8]. The attack results for [7] and [8] are given in Sections III-C and III-D, respectively.

E. Attack on Memristor-Based Obfuscation [6]

1) Working: A memristor-based voltage divider in [6] tunes the bulk terminals of the differential pair in the sense amplifier. This tuning is required to cancel the output offset voltage for zero input differential voltage. The voltage divider consists of two memristor crossbars. Each crossbar is constructed using

TABLE II

SMT-BASED ATTACK IS EFFECTIVE ON THE ANALOG LOCKING TECHNIQUES IN [6]–[9] AND [14]. THE EQUATIONS SHOWN HERE ARE BASED ON THE CIRCUIT WHICH IS PROTECTED. THE EQUATIONS MAY VARY BASED ON THE CIRCUIT TOPOLOGY

Defense technique	Bias affected	Circuit parameter affected (p)	Attack equations		
			Equation (A): The value of obfuscated component ($\tilde{y_v}$) as a function of key	Equation (B): Bias input (z_{ob_comp}) as a function of $\tilde{y_v}$	Equation (C): Bias input (z_{spec}) as a function of p
[7], [8], [14]	I_B, V_B	g_m, BW, f_{osc}, Q, f_c	$\tilde{y_v} = \sum_{i=1}^{n} x_{i_v} q_k$ $\tilde{y_v} = R = \frac{1}{g_m} = \frac{1}{\sqrt{2\mu C_{ox}\left(\frac{W}{L}\right) I_D}}$	$I_B = \tilde{y_v} \times I_{ref}$ $V_B = \frac{\tilde{y_{2_v}} \times V_{ref}}{\tilde{y_{1_v}} + \tilde{y_{2_V}}}$	$I_B = \frac{g_m^2}{\mu C_{ox} \frac{W}{L}}$
[6]	V_{PROG}	V_{BB}	$\tilde{y_{1_v}} = R_{UPPER} = \left(\sum_{i=1}^{U} \frac{q_k}{x_{i_v}} \right)^{-1}$ $\tilde{y_{2_v}} = R_{LOWER} = \left(\sum_{i=1}^{L} \frac{q_k}{x_{i_v}} \right)^{-1}$	$V_{PROG} = \frac{\tilde{y_{2_v}} \times A \times V_{PP}}{\tilde{y_{1_v}} + \tilde{y_{2_v}}}$	$V_{PROG} = \frac{\omega_{PT} \times R_M}{2\pi\rho} \times \left(\sqrt{\frac{\varphi}{\gamma}} \right)$ here, $R_M = \frac{V_{BB} \times R_1}{V_{DD} - V_{BB}}$
[9]	–	g_m, BW, f_{osc}, Q, f_c	$\tilde{y} = \left(\frac{W}{L}\right)_{camouflaged} = \sum_{i=1}^{3} x_i k_i \times \left(\frac{W}{L}\right)_{NVT}$	–	$g_m = \sqrt{2\mu C_{ox}\left(\frac{W}{L}\right)_{camouflaged} I_D}$

an array of memristors, as shown in Fig. 4. The key determines the connectivity among these memristors and the effective resistances of the upper (R_{UPPER}) and lower (R_{LOWER}) memristor arrays. Applying the correct key configures the resistivity of the crossbars to provide the required body-bias voltage (V_{BB}). This V_{BB} helps in the offset voltage compensation. An incorrect key provides an undesired V_{BB}, which does not compensate for the offset voltage. This offset voltage affects the sensitivity and reliability of the sense amplifiers.

2) Attack Methodology: The security of this technique lies in the preprogrammed memristor crossbar. It can be compromised if the attacker finds: 1) the value of the preprogrammed resistance in each crossbar and 2) the connectivity among the memristors, controlled by the key inputs.

1) *Obfuscated Component:* The effective resistance of the obfuscated upper and lower memristor crossbars are $\tilde{y_{1_v}}$ and $\tilde{y_{2_v}}$, respectively. Here, $\tilde{y_{1_v}} = R_{\text{UPPER}} = (\sum_{i=1}^{U}(q_k/x_{i_v}))^{-1}$, where $k \in \{1, 2, \ldots, m\}, \forall i$ and $\tilde{y_{2_v}} = R_{\text{LOWER}} = (\sum_{i=1}^{L}(q_k/x_{i_v}))^{-1}$, where $k \in \{1, 2, \ldots, m\}, \forall i$. Here, $\mathbf{q} = (q_1, q_2, \ldots, q_m)$ is an m-bit key. This key is shared among both the crossbars. U and L are the number of memristors connected in parallel in the upper and lower crossbars, respectively. x_{i_v} is the preprogramed memresistance value of memristor i. The attacker can deduce the value of U and L from the layout or reverse-engineered netlist. From the circuit specification, he/she knows the minimum and maximum resistance values of the memristors.
2) *Equations Linking the Bias z_{ob_comp} With Obfuscated Resistivities $\tilde{y_{1_v}}$ and $\tilde{y_{2_v}}$ of the Crossbars:* The voltage divider generates the necessary programming voltage, V_{PROG} for the memristor M. The equation linking the bias V_{PROG} with $\tilde{y_{1_v}}$ and $\tilde{y_{2_v}}$ is $z_{\text{ob\_comp}} = V_{\text{PROG}} = \psi(\tilde{y_{1_v}}, \tilde{y_{2_v}})$. Here, $\psi(\tilde{y_{1_v}}, \tilde{y_{2_v}}) = (\tilde{y_{2_v}}/\tilde{y_{1_v}} + \tilde{y_{2_v}}) \times AV_{\text{PP}} = ((\sum_{i=1}^{L}(q_k/x_{i_v}))^{-1})/((\sum_{i=1}^{U}(q_k/x_{i_v}))^{-1} + (\sum_{i=1}^{L}(q_k/x_{i_v}))^{-1}) \times AV_{PP}$. Here, A is the amplifier's gain and V_{PP} is the peak-peak voltage. V_{PROG} and V_{PP} can be derived from the circuit specification. The amplifier's gain can be computed from the layout or the reverse-engineered netlist.
3) *Equation Linking the Bias z_{spec} With Circuit Parameter p:* The equations connecting V_{PROG} with V_{BB} are $V_{\text{BB}} = (R_M/R_1 + R_M) \times V_{\text{DD}}$, where $R_M = ((\gamma/\varphi))^{1/2} \times \rho \times V_{\text{PROG}}(2\pi/\omega_{\text{PT}})$. Combining the equations gives $z_{\text{spec}} = V_{\text{PROG}} = \theta(p) = \theta(V_{\text{BB}})$. Here, V_{DD} is the supply voltage, R_1 is the fixed resistor in the voltage divider, and R_M is the effective resistance of the memristor M. γ is a constant depending on device parameters such as carrier mobility and device thickness, φ is the flux, ρ is the duty cycle, and ω_{PT} is the frequency of programming pulse. The values of V_B, V_{DD}, φ, γ, ρ, ω_{PT}, and R_1 are available in the circuit specification. The resistivity range with which the memristors can be preprogrammed is (R_{min}, R_{max}), where R_{min} and R_{max} are the minimum and maximum resistivity of the memristor M. The attacker can obtain the value of R_{min} and R_{max} from the circuit specification.
4) *Putting It All Together:* Solving these equations gives the correct key and the resistance of each memristor. These equations are consolidated in Table II as equations (A), (B), and (C). There can be more than one correct key that gives the same V_{BB} due to the memristor array configuration. Hence, the SMT solver is called only once to determine one correct key and one set of memristors' resistance values.

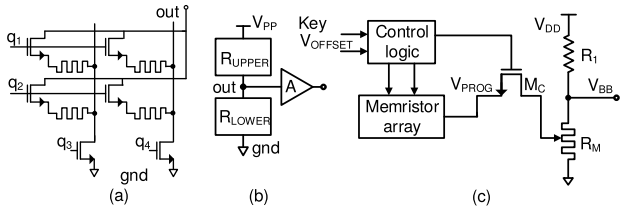

Fig. 4. Memristor-based obfuscation technique [6]. (a) Memristor crossbar architecture used in the voltage divider circuit. (b) Output of the memristor-based voltage divider is amplified by a factor of A. (c) For a non-zero V_{OFFSET}, the control logic turns on the transistor M_c. This passes the V_{PROG} generated by the memristor array to program the memristor R_M.

F. Attack on Analog Performance Locking [10]

1) Working: In this technique, a trained ANN provides precise I_B to OTA. This I_B is required for the proper operation of the OTA. The ANN's core shown in Fig. 5(a) consists of an $n \times n$ array of synapses (S) and neurons (N). Here, n is the number of rows and columns in the ANN. The first and the last row of synapses are called the input and output layer, respectively. The rows in between them are the hidden layers. Each synapse implements an analog multiplier. Likewise, each of the neurons implements a non-linear activation function,

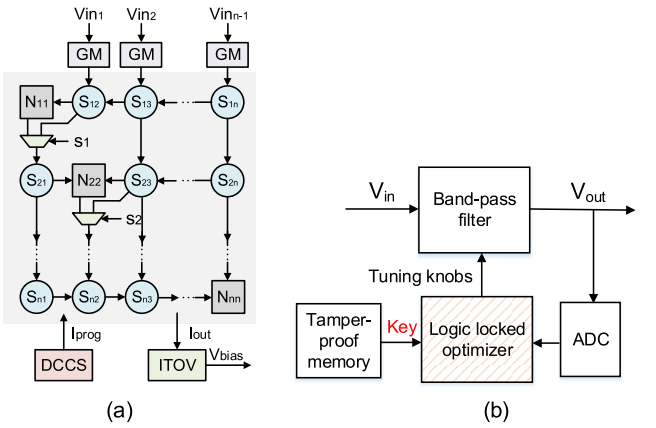

Fig. 5. (a) ANN along with the input differential transconductance (GM), DCCS, and current to voltage converter (ITOV). The core of the neural network consists of the neurons (N) and synapses (S). (b) Logic locking of the BPF circuit. Only by applying the correct key, the optimizer sets the correct resistor value by considering the effect of process variation [11].

e.g., tanh. The network is trained in such a way that for a given set of input voltages, it determines the weights of each synapse to generate the required V_B. Also, it is possible to train the ANN to provide the same V_B for different inputs.

This attack mathematically models the ANN using the SMT formulations. The synapse outputs the product of the inputs along with the weight associated with it. S_{ij} is the synapse output, where i and j are the row and column number of the synapse considered, respectively. Sw_{ij} is the weight associated with the synapse S_{ij}. $\text{IN}_{1_{ij}}$ and $\text{IN}_{2_{ij}}$ are the two inputs to S_{ij}. Then, S_{ij} is modeled as, $S_{ij} = Sw_{ij} \times \text{IN}_{1_{ij}} \times \text{IN}_{2_{ij}}, i \neq j$ where $i, j \in (1, \ldots, n)$. The output of the neurons (N_{ij}), which forms the diagonal elements of the ANN matrix, depends on the select signal s_i, and is given by

$$N_{ij} = \begin{cases} \tanh(S_{i(j+1)}), & \text{if } s_i = 1 \wedge i = j = 1 \\ \tanh(S_{i(j-1)}), & \text{if } s_i = 1 \wedge i = j = n \\ \tanh(S_{i(j-1)} \times S_{i(j+1)}), & \text{if } s_i = 1 \wedge \text{otherwise} \\ S_{i(j+1)}, & \text{if } s_i = 0. \end{cases}$$

The inputs to the synapses in the input layer ($i = 1$) are given by $\text{IN}_{1_{1j}} = \text{IN}_{i-1}$ and $\text{IN}_{2_{ij}} = S_{1(j+1)}$ and for other layers, the inputs are given by the following equations.

$$\text{IN}_{1_{ij}} = \begin{cases} S_{(i-1)j}, & \text{if } j \neq i-1 \\ N_{(i-1)j}, & \text{if } j = i-1 \end{cases}$$

$$\text{IN}_{2_{ij}} = \begin{cases} 1, & \text{if } j = 1 \vee j = n \\ S_{i(j-1)}, & \text{if } j < i \\ S_{i(j+1)}, & \text{if } j > i. \end{cases}$$

2) Attack Methodology: One method to attack this technique is to remove the ANN and replace it with the bias circuit. However, as stated in [10], this defense technique claims resilience only against illegitimate access to the chips. Hence, the attacker cannot remove the ANN; rather, he/she should program the synapses and neurons with correct weights to produce the desired I_B/V_B. The SMT formulations for the ANN, the differential transconductor, and the current-to-voltage converter are fed to the SMT solver along with the required V_B range. This bias range is essential for the proper operation of the OTA. The solver returns the input voltages to the ANN, weights associated with each synapse, the type of neuron (buffer or tanh activation function), and the value

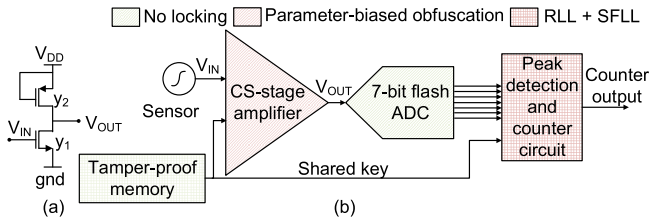

Fig. 6. (a) CS amplifier. (b) Shared dependencies lock [14]. The CS-stage amplifier is locked using parameter-biasing obfuscation technique [8]. The digital circuit is locked using RLL and SFLL [3].

of V_B. The attacker can procure an off-the-shelf digitally controlled current source (DCCS) to program the new weights into the synapses. The input voltages returned by the attack are fed to the input layer synapses. The ANN thus produces the required I_B or V_B, thereby rendering the attack successful. More than one correct configuration gives the same V_B due to the neural network topology [10]. Hence, the SMT solver is called only once to determine one correct configuration that provides the necessary bias.

G. Attack on Camouflaged Analog IPs [9]

1) Working: The threshold voltage required to switch on a transistor is camouflaged by fabricating the transistor with different dopant concentrations. The threat model considers a trusted foundry and an untrusted end-user. This means the attacker does not have access to the foundry, where he/she can determine the V_{th} type from the layout. TSMC fabricates the transistors in 180-nm technology with nominal-V_{th} (NVT), medium-V_{th} (MVT), and native-V_{th} (NaVT). In this technique, few of the NVT transistors are replaced by MVT and NaVT transistors [9]. The functionality is maintained by resizing the replaced transistors. Thus, each camouflaged transistor can be modeled as

$$\left(\frac{W}{L}\right)_i = x_i \times \left(\frac{W}{L}\right)_{\text{NVT}}. \tag{5}$$

Here, $i \in \{\text{NVT}, \text{MVT}, \text{NaVT}\}$.

2) Attack Methodology: The attacker can transform individual camouflaged transistors to logic-locked transistors [32]

$$\left(\frac{W}{L}\right)_{\text{camouflaged}} = \begin{cases} \left(\frac{W}{L}\right)_{\text{NVT}}, & \text{if } q_1 = 1 \\ \left(\frac{W}{L}\right)_{\text{MVT}}, & \text{if } q_2 = 1 \\ \left(\frac{W}{L}\right)_{\text{NaVT}}, & \text{if } q_3 = 1. \end{cases} \tag{6}$$

Here, $q_1, q_2, q_3 \in \{0, 1\}$ are the key-bits controlling transistors of type NVT, MVT, and NaVT, respectively. Also, $q_1 + q_2 + q_3 = 1$, as each camouflaged transistor can be only one of the three types. The attacker performs the SMT-based attack on the transformed logic-locked circuit as follows.

1) *Obfuscated Component:* The transistors in the design are the obfuscated components, as their V_{th} type is unknown to the attacker. From (5) and (6), the aspect ratio of the camouflaged transistor $\tilde{y} = \phi(x, q)$ is

$$\tilde{y} = \frac{W}{L}_{\text{camouflaged}} = \left(\sum_{\forall i} x_i q_i\right) \times \left(\frac{W}{L}\right)_{\text{NVT}}. \tag{7}$$

Here, $i \in \{\text{NVT}, \text{MVT}, \text{NaVT}\}$ and $q_i \in \{0, 1\}$ is the key-bit controlling transistor of type i.

2) *Linking $\tilde{y}$ to the Circuit Specification of the Locked Analog IC:* Unlike techniques that obfuscate the bias circuits, analog camouflaging obfuscates the transistors that affect the circuit specification. Considering the fourth-order Gm-C BPF, the transconductance is given by $g_m = (2\mu C_{\text{ox}}\tilde{y}I_D)^{1/2}$ where $I_D = I_B/2$. Solving equations (A) and (C) for [9] in Table II gives the required key for the precise operation of the BPF. From this key, the attacker determines the variant of the transistor.

H. Attack on AMSlock [11]

The AMSlock and Mixlock [12] are the same locking technique, as both use SFLL-HD0/h [3] to lock the digital section of the AMS circuit. The digital optimizer in AMSlock and the digital decimation filter in the Mixlock are locked using SFLL-HD^0 or SFLL-HD^h [3]. These locked circuits receive inputs from the ADC in AMSlock and from the $\Delta\Sigma$ ADC in Mixlock. In [11], the ADC is fed by the analog circuits, such as BPF, *LC* oscillator, or triangular waveform generator (TWG). Whereas, in [12], the $\Delta\Sigma$ ADC is fed with the audio input, which has to be modulated. The defender chooses the PIPs by analyzing the inputs from the ADC or the $\Delta\Sigma$ ADC. Therefore, the attacker determines the PIPs by simulating the analog circuit and ADC in [11] and analyzing the audio signals sent to the $\Delta\Sigma$ ADC in [12]. He/she then uses SAT formulations to determine the correct key. The following section explains our attack on AMSlock [11]. The only difference between these two techniques is the circuit over which the defense is implemented. However, the underlying defense algorithm remains the same. Therefore, the attack which we have shown on AMS lock can break Mixlock too.

1) Working: The purely digital optimizer controls the value of the passive components, such as R and C, of the analog circuit-under-protection. This optimizer is locked using the SFLL [3]. For the correct key, it chooses the correct value of these components via the tuning knobs, considering the impact of process variations. However, for an incorrect key, the optimizer does not consider the impact of these variations. Hence, incorrect values are chosen, leading to circuit malfunction.

2) Attack Methodology: This defense thwarts an attacker from overproducing the chip but cannot thwart him/her from modifying the layout of the design and pirate (IP piracy). Hence, the attacker can assume access only to the layout but cannot modify the same. This design has 1024 tuning knob settings corresponding to 1024 unique resistor settings. These tuning knobs are internal to the chip and are not available as top-level ports. Adding to this, the optimal settings of the tuning knobs vary chip to chip due to the process variations. Therefore, the attacker cannot simulate the analog circuit-under-protection for all the tuning knob settings to determine the correct settings as it changes chip to chip. Only the optimizer can control the tuning knob settings. As the attacker cannot modify the tuning knob settings directly, it is necessary to determine the correct key to unlock the locked optimizer.

1) *Obfuscated Component:* To reduce the impact of PVT variations and mismatch, passive components, such as R and C, are often implemented as banks of elements to enable calibration. The correct value of the passive components is chosen by the locked optimizer and cannot be computed by analyzing the netlist. Hence, we identify this component as the obfuscated component.
2) *Equation Linking the Obfuscated Component and the Key Inputs to the Optimizer:* In the BPF circuit, the resistors R_1 and R_2 are the obfuscated components. The correct value of these resistors is chosen by two 5-bit tuning knobs, which are controlled by the locked optimizer. Each of the 1024 tuning knob settings corresponds to a unique resistor setting. The output response of the analog circuit can be determined from the transfer function given by $H(s) = (s/(R_1C)/s^2 + s/(R_1C) + 1/(R_2^2C))$. Here, C is the fixed capacitor. The attacker can simulate the output response of the circuit for unique resistor settings, via transistor-level simulations. As there are only 1024 unique tuning knob settings, the analog circuit can have 1024 unique output responses. The output response is digitized using the ADC. These digitized output responses are the INs that are fed to the locked optimizer. The optimizer chooses the tuning knobs based on these inputs. These INs are required to determine the SAT formulations for the attack.
3) *Breaking SFLL-HD^h [3]:* The SFLL-HD^h technique can have more than one correct key for a PIP when $h > 0$ [3]. If the attacker finds one key that ensures the correct output for all 1024 PIPs, it can be used to unlock the overproduced chip.
4) *Finding PIPs:* The attacker can determine the PIPs in the 1024 INs with the help of oracle. The entire AMS chip loaded with the correct key constitutes the oracle. Only the input and output ports of the analog circuit-under-protection in this chip are available to the attacker. Hence, the attacker has to simulate the analog circuit for different tuning knob settings to determine the INs to the locked optimizer. The signal generator gives the required input to the oracle [33], and he/she can observe the oracle's response on the output port. If the locked optimizer gives an incorrect output for an IN, then it is a PIP. Otherwise, it is not a PIP. Hence, if there are p PIPs out of 1024, then the remaining n patterns $(1024 - p)$ are unprotected IPs.
5) *SAT Formulations:* Along with the locked netlist (N_{locked}) and the HD used by the defender, the following are the other constraints added to the SAT formulations.
   a) The output response (O) of the analog circuit corresponding to each PIP is found using the oracle. The corresponding constraint is given by $(\text{PIP}_1 \Rightarrow O_1) \wedge (\text{PIP}_2 \Rightarrow O_2) \wedge \cdots \wedge (\text{PIP}_p \Rightarrow O_p)$.
   b) HD between each PIP and the key (K) must be equal to h, $\sum_{i=1}^{p} \wedge(\text{HD}(\text{PIP}_i, K) = h)$.
   c) HD between other n INs and K should not be equal to h, $\sum_{i=1}^{n} \wedge(\text{HD}(\text{IN}_i, K) \neq h$.

   Combining all the abovementioned constraints gives

$$\begin{aligned} N_{\text{locked}} \; &\wedge \text{PIP}_p \wedge O_p \wedge (\text{PIP}_1 \Rightarrow O_1) \\ &\wedge (\text{PIP}_2 \Rightarrow O_2) \cdots \wedge (\text{PIP}_p \Rightarrow O_p) \\ &\wedge \sum_{i=1}^{p} \wedge(\text{HD}(\text{PIP}_i, K) = h) \\ &\wedge \sum_{i=1}^{n} \wedge(\text{HD}(\text{IP}_i, K) \neq h). \end{aligned} \tag{8}$$

Equation (8) helps in determining the correct key.

I. Attack on Shared Dependencies [14]

1) Working: This technique improves the resiliency against IP piracy and overproduction by locking the analog and

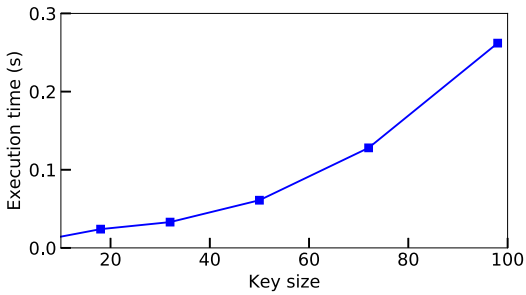

Fig. 7. (Successful) attack time for increasing key size in memristor-based protection technique [6].

digital parts of an AMS circuit. The AMS circuit-under-protection consists of a sensor, a common-source (CS) amplifier, a 7-bit flash ADC, a peak detection, and a counter circuit. The CS amplifier, the peak detection circuit, and the counter are locked using parameter-biasing obfuscation [8], SFLL-HD^0 [3], and random logic locking (RLL) [2], respectively. A set of N transistors in parallel, $\{x_{11}, x_{12}, \ldots, x_{1N}\}$, replaces the transistor y_1 in Fig. 6. Similarly, y_2 is replaced by $\{x_{21}, x_{22}, \ldots, x_{2N}\}$. The single transistor getting replaced with N transistors is similar to the replacement shown in Fig. 3(a) and (b) illustrating [1]. This is because combinational lock [7] and parameter-biasing obfuscation [8] are similar and replace a single transistor with multiple transistors. The effective width of y_1 and y_2 is controlled by key ($\mathbf{q}$) of size $2 \times N$. Each of the digital- and analog-locked sections has a dedicated part of the whole key.

2) Attack Methodology: The attacker has to find the correct key required to unlock the analog and digital parts of the AMS circuit. This technique's threat model assumes an untrusted foundry and an untrusted end-user. Hence, he/she can access the layout, the oracle, and the circuit specification. The attacker targets the analog and digital locks separately.

3) Breaking the Digital Lock: The output of the ADC is the input to the peak detection circuit locked using SFLL-HD^0. This circuit sets the peak detection signal to one when the ADC output is maximum and resets to zero when the ADC output falls below the maximum value. The PIP corresponds to the maximum ADC value. In SFLL-HD^0, as the key is equal to the PIP, the attacker can unlock the locked peak detection circuit. He/she could verify the correctness of the key found, using the SAT formulations provided in Section II-H. The SAT attack [15] can unlock the counter circuit. Thus, the attacker has determined the key to unlock the digital part of the AMS circuit even without unlocking the locked analog circuit.

4) Breaking the Analog Lock: The following SMT formulations help to determine the key to unlock the analog circuit.

1) *Obfuscated Component:* The effective aspect ratio of the obfuscated transistors y_1 and y_2 are y_{1_v} and y_{2_v}, respectively. Here, $\tilde{y_{1_v}} = (W/L)_{y_1} = \sum_{i=1}^{N} q_k \times x_{1i_v}$, where $k \in \{1, 2, \ldots, N\}, \forall i.\ \tilde{y_{2_v}} = (W/L)_{y_2} = \sum_{i=N+1}^{2\times N} q_k \times x_{2i_v}$, where $k \in \{N+1, N+2, \ldots, 2\times N\}, \forall i.\ x_{1i_v}$ and x_{2i_v} are the aspect ratios of transistors x_{1i} and x_{2i}, respectively. The attacker knows the value of N from the layout.
2) *Equations Linking the Gain z_{ob_comp} With Obfuscated Aspect Ratios y_{1_v} and y_{2_v} of the Transistors:* The CS amplifier generates the necessary analog input to the ADC. The equation linking the amplifier's gain (G_{CS}) with y_{1_v} and y_{2_v} is $z_{\text{ob\_comp}} = G_{\text{CS}} = -(1/1+\eta)(\tilde{y_{1_v}}/\tilde{y_{2_v}})^{1/2}$. Here, η is the backgate transconductance available in PDK.
3) *Equation Linking the Gain z_{spec} With Circuit Parameter:* The gain of the CS amplifier is in the specification.
4) *Putting It All Together:* Solving the abovementioned equations gives the correct key and hence the effective aspect ratios of the obfuscated transistors.

Using the abovementioned formulations, the attacker can find the correct key to unlock the AMS circuit. Section III provides the attack results on this defense technique.

III. Attack Results

A. Estimation of Unlocked Circuit Performance

Once the attacker determines the key using the SMT/SAT solvers, this key is applied to the locked circuit. As the attacker has the layout or the reverse-engineered netlist, we have considered access to layout to demonstrate our attacks. In a real attack, the output response of the unlocked circuit is measured using an oscilloscope. This response is then compared with the oracle's response to check the correctness of the deduced key. In this work, the simulations are based on a transistor-level netlist from the layout available in the foundry. The attacker simulates this netlist, which does not have any deviation from the defender's design. The extracted netlist comprises the analog circuit-under-protection and the locked bias circuitry. He/she also has access to the reverse-engineered netlist, as mentioned in Table I. As RE is an imprecise and expensive process, the extracted netlist may not be accurate with respect to the dimensions of the transistors and passive components. As there can be differences in the output responses of the circuit via simulations of the extracted netlist, the impact of imprecise RE on our attack is included in Section IV.

B. Experimental Setup

The transistor-level schematics of the analog and AMS circuits used to evaluate the proposed attack are based on the IBM 180-nm technology library using Cadence Virtuoso. Our attack is demonstrated in the following:

1) the OTA, fourth-order Gm-C BPF, quadrature oscillator, LC oscillator, and TWG used in class-D amplifiers locked using combinational locks [7] and parameter-biasing obfuscation [8];
2) the BPF, LNA, and low-dropout (LDO) locked using AMS lock [11];
3) the OTA locked using analog performance locking [10].

We use iSAT3 solver [31] and pycosat [34] to solve all the SMT-based attack formulations and SAT-based attack formulations, respectively. These experiments are run on x86 architecture, 64-bit Intel Xeon CPU processor with 16 cores per socket and two threads per core.

C. Attack Results on Combinational Locks [7]

1) Attack and Defense Time: In the combinational locking [7], the equations relating the key inputs, $I_{\text{ref}}/V_{\text{ref}}$, I_B/V_B, and the transistor sizes are generated by the SMT solver. The defense time is the time the solver takes to generate these equations. Similarly, the attack time in Table III corresponds to the time the SMT solver takes to solve the attack equations and determine the unique key required to unlock the current mirrors. The time taken to formulate the attack equations is not taken into account, as it is done manually. Fig. 8 illustrates the attack and defense times on a combinational

TABLE III

COMBINATIONAL LOCK [7] AND PARAMETER-BIASING OBFUSCATION [8] DEFENSE AND THE PROPOSED ATTACK RESULTS. FOR EACH INCORRECT KEY, THE CONFIGURABLE CURRENT MIRROR IN THE DEFENSE SETUP IS CONFIGURED SUCH THAT I_B IS EITHER $<(1-\Delta)I_B(I_{B_{\min}})$ OR $>(1+\theta)I_B(I_{B_{\max}})$, WHERE $\Delta = 0.8$ IS THE LOWER BOUND AND $\theta = 3$ IS THE UPPER BOUND ON I_B. INSTANCE (INST)

Benchmark	Circuit details							Defense				Attack								
				[7]		[8]		[7]			[8]	[7]							[8]	
	# CM inst	I_{ref} (μA)	I_B (μA)	Matrix size	Key size	Matrix size	Key size	$I_{B_{min}}$ (μA)	$I_{B_{max}}$ (μA)	Time reqd. (s)	Time reqd. (s)	Individual CM I (μA)					Time reqd. (s)	# of calls	Time reqd. (s)	# of calls
												I_{B1}	I_{B2}	I_{B3}	I_{B4}	I_{B5}				
BPF	1	0.1	2	2×6	84	–	512	0.4	8	1351		2	2	2	2	2	0.829	6	0.158	1
	2	2	38	2×6		1×64		7.6	152	1080	3	38	38	38	38	38				
	3	0.1	2	2×6		–		0.4	8	1351	–	101	2	2	2	2				
	4	2	112	2×6		1×64		22.4	448	9893	2	606	10	10	10	112				
	5	1	20	2×5		1×64		4	80	8	9	20	20	20	20	20				
	6	1	38	2×5		1×64		7.6	152	1249	3	7	420	425	425	38				
	7	0.1	4	2×6		–		0.8	16	15	–	4	4	4	4	4				
	8	4	112	2×5		1×64		22.4	448	1260	7	112	112	112	112	112				
	9	1	56	2×6		1×64		11.2	224	27300	4	56	56	56	56	56				
	10	0.5	20	2×6		1×64		4	80	16980	3	20	20	20	20	20				
	11	0.1	2	2×5		–		0.4	8	8	–	2	2	2	2	2				
	12	2	56	2×7		1×64		11.2	224	18060	19	56	56	56	56	56				
LC osc.	1	2	30	2×6	14	1×256	512	6	120	240	1983	20					0.092	2	0.184	1
	2	30	270	2×7		1×256		54	1080	5940	270	270								
TWG	1	2	100	2×6	15	1×256	512	20	400	86400	170	100					0.095	2	0.212	1
	2	2	100	2×7		1×256		20	400	25200	1079	100								
Quad osc.	1	4	280	4×3	12	1×256	512	56	1120	76800	413	280					0.094	2	0.314	1
	2	4	556	4×4		1×256		111.2	2224	50	35	556								

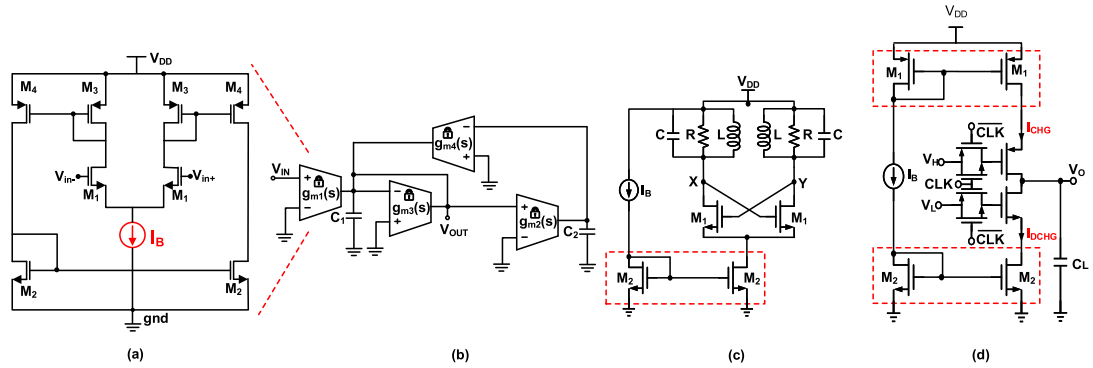

Fig. 9. (a) OTA with the locked current bias indicated in red. (b) Second-order Gm-C BPF [7] constructed using four locked OTAs. (c) The current mirror producing the required bias to the *LC* oscillator is locked using combinational lock [7], as shown by red dashed box. (d) Schematics of TWG. The two current mirrors generating I_{CHG} and I_{DCHG} are locked using combinational lock [7] as indicated in red dashed boxes.

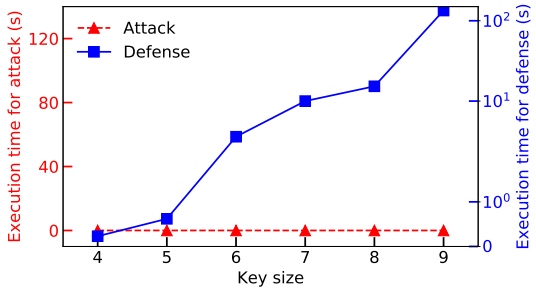

Fig. 8. For a constant number of transistor switches, the defense time shows an exponential pattern with increasing key size, whereas the attack requires a constant time of 0.01 s.

locked OTA for increasing key sizes. It is evident from the figure that the attack time is constant compared to the defense time. The attack time does not depend on the key and matrix size. In the combinational lock [7], the SMT constraints for the defense, size the transistors such that only one key gives the required I_B. For all other key vectors, it gives an incorrect I_B value. Hence, it requires the enumeration of each key combination in the input constraints to the SMT solver. The defense time also depends on other factors such as the key size, I_{ref}, I_B, Δ, and θ, as explained in [7].

TABLE IV

ATTACK RESULTS ON AMS LOCKED CIRCUITS [11]. *h* DENOTES HD

Benchmark	Hamming distance = 0				Hamming distance = h				
	Key size	Effective key size	Time taken (s)	# of calls	h	Key size	Effective key size	Time taken (s)	# of calls
BPF	87	87	1.26	1	7	220	177	90	1
	112	112	1.9	1	15		144	58	1
	220	220	1.5	1	20		126	11507	1
LNA	81	81	1.74	1	4	154	129	35	1
	84	84	1.49	1	11		99	55	1
	154	154	1.78	1	21		68	444	1
LDO	109	109	2.38	1	7	234	191	84	1
	135	135	2.52	1	14		160	103	1
	234	234	2.96	1	28		114	85140	1

As the number of SMT constraints increases exponentially with respect to the key size, the time taken also increases exponentially, as shown in Fig. 8. The results are illustrated on smaller key sizes to show the defense and attack time trends. The maximum key size on which we execute our attack is dependent on the locking technique used. Hence, we could not increase the key size to more than 15 bits for circuits that require only two current mirrors such as *LC* oscillator, TWG, and quadrature oscillator. However, this attack can successfully determine the keys of sizes 80–512 bits, as shown in Tables III–V. In this attack, the number of calls to the SMT solver is equal to the number of keys that satisfy the attack equations plus one. Each call returns one key, and the last call

TABLE V

ATTACK INFORMATION IS TABULATED FOR SHARED DEPENDENCE [14] AND ANALOG PERFORMANCE LOCKING [10].

Shared dependency [14]							ANN [10]		
Total key size	Analog			Digital			ANN size	Attack time (mins)	Bias voltage
	Key size	Defense time (s)	Attack time (s)	RLL key	SFLL key	Attack time (s)			
160	80	0.642	0.101	40	40	0.03	20×20	3	0.705
200	100	0.943	0.113	50	50	0.035	30×30	18	0.425
240	120	1.268	0.122	60	60	0.06	40×40	134	0.872
280	140	1.622	0.127	70	70	0.1	50×50	1243	0.4
320	160	2.183	0.127	80	80	0.1	60×60	3383	0.11

returns no solution once all the keys are found. The number of circuit simulations is equal to the number of keys that satisfy the attack equations.

2) Fourth-Order Gm-C BPF: It has two second-order BPFs shown in Fig. 9(b) in cascade. It is characterized by $f_c = 250$ kHz, BW $= 150$ kHz, and the amplitude gain $= 0$ dB. The leading second-order BPF is implemented with the capacitance $C_{11} = C_{12} = 78.95$ pF, $f_{c1} = 201.6$ kHz, and $\text{BW}_1 = 83.6$ kHz. The succeeding one has $C_{21} = C_{22} = 51.34$ pF, $f_{c2} = 310$ kHz, and $\text{BW}_2 = 128.5$ kHz. Each of the second-order BPF contains four OTAs. The CCM feeds each of the OTAs with the required I_B, as shown in Fig. 3. Table III lists the size of the transistor switch matrix, replacing the mirroring transistor in each current mirror.

Here, C_1 and C_2 are the capacitances. g_{m1}, g_{m2}, g_{m3}, and g_{m4} are the transconductance of OTAs 1, 2, 3, and 4, respectively. From the attack equations, we can infer that f_c, G_{BPF}, and Q are dependent on the transconductance g_m. Using the circuit specification and the attack equations, the attacker can calculate g_{m1} and g_{m3}. Hence, from the equation, $g_m = (2\mu C_{\text{ox}}(W/L)I_D)^{1/2}$, where $I_D = I_B/2$, I_{B_1} and I_{B_3} corresponding to OTAs 1 and 3 can be calculated. Though he/she does not have the information to calculate I_{B_2} and I_{B_4} individually, he/she can calculate the product of these bias currents via the product of transconductances ($g_{m2} \times g_{m4}$). Therefore, I_{B_1}, I_{B_3}, and $I_{B_2} \times I_{B_4}$ are provided to the SMT solver along with equations (A) and (B) listed in Table II for each of the second-order BPF. The equations corresponding to all the CCMs are solved together.

Our attack obtains the key even without knowing the ranges for all the bias currents in 0.829 s, as shown in Table III for [7]. As f_c and Q depend on $g_{m_2}g_{m_4}$ rather than individual g_m, the attack equations return five possible keys (which include the correct key $\mathbf{q}^*$ from defense). The attacker hence reduces the key search space from 2^{84} to five using this attack. The BPF is simulated for the five different key inputs reported by the attack formulation in Fig. 10(a). Key 1 shows the response of the correct key, where $f_c = 250$ KHz with an amplitude gain of 0 dB. As analog circuits are sensitive to biasing conditions, we could see a degradation of 2 dB near the center frequency for keys 2–5. As mentioned in Section II-A, the attacker, as an end-user, has access to the unlocked chip. He/she can compare the output responses of this circuit for the deduced keys with the oracle's response. This comparison helps in finding the correct key. Here, the analog circuits are not simulated for all the key combinations. They are simulated only for the keys reported by the SMT solver after solving the attack equations. Hence, even if the time taken to simulate the analog circuits increases for larger circuits and larger key sizes, its impact on the overall attack time is limited. However, note that this attack can prune a large set of keys into a handful ones, thereby reducing the number of simulations significantly. This process can be automated using the analog simulation tools offered by companies, such as Cadence, Synopsys, and Mentor Graphics.

3) LC Oscillator: produces a clock signal with the oscillation frequency $f_{\text{osc}} = 1/2\pi\sqrt{LC}$ [28], where L is the value of the inductor and C is the capacitance. An LC oscillator is designed with $f_{\text{osc}} = 2$ GHz and the amplitude of the output oscillation $V_o = 2.3V$. The defender protects V_o, which is equal to $4I_B(w_{\text{osc}}L)^2/\pi R_s$, by locking the current mirror that generates I_B, as shown in Fig. 9(c). Table III lists the size of the switch matrix for the individual CCMs and the time taken to find the sizes of the switch transistors for [7]. Here, $\omega_{\text{osc}} = 2\pi f_{\text{osc}}$, and R_s is the series resistance. To obtain the correct key, the attacker gathers the values of L, C, and R_s using the locked netlist and PDK. He/she determines the value of V_o and w_{osc} from the circuit specification.

As the CCMs are in cascade, the attacker also includes $I_{\text{ref}_2} = I_{B_1}$ in the SMT formulations. Solving these equations along with the corresponding equations in Table II gives the correct key in 92 ms. The values of the bias currents from the unlocked CCMs are equal to the original circuit's values, making the chip functional. Fig. 10(b) shows the simulation results indicating the frequency of operation and output voltage swing for the identified key. These values match with the circuit specification, thereby demonstrating a successful attack.

4) Triangular Waveform Generator: In class-D amplifiers, the output of the pulsewidth modulated signal depends on the frequency of charging (I_{CHG}) and discharging (I_{dCHG}) ramps of the carrier signal generated by the TWG [29] shown in Fig. 9(d). The carrier signal has the frequency of $f_{\text{TRI}} = 500$ KHz with the output capacitor, $C_{\text{TRI}} = 100$ pF, maximum voltage $V_H = 500$ mV, and minimum voltage $V_L = -500$ mV. As this carrier signal significantly impacts the total harmonic distortion of the amplifier, the defender locks the current mirrors of the TWG with a 15-bit key. From the circuit specification, the attacker can determine T_{TRI}, C_{TRI}, V_H, and V_L. Thus, he/she can formulate the equation to calculate the charging (I_{Chg}) and discharging (I_{DChg}) currents: $T_{\text{TRI}} = C_{\text{TRI}}(V_H - V_L)((1/I_{\text{Chg}}) - (1/I_{\text{DChg}}))$. As shown in Table III, our attack takes only 95 ms. Fig. 10(c) shows the time period of the triangular waveform for the key found by the attack. It is equal to 2 μs, which is the desired time period. Thus, the key returned by the attack is indeed valid.

5) Quadrature Oscillator: The center frequency of the oscillations is given by $f_c = 2.34$ MHz. The oscillator consists of two OTAs. The transconductance (g_m) of the OTAs is set to 1 mA/V. There are two current mirrors locked using [7] with only a 12-bit key, due to the SMT constraints in defense explained Section III-C. As illustrated in Table III, the attacker can find the key in 95 ms.

D. Attack Results on Parameter-Biasing Obfuscation [8]

The defense and attack time calculations are similar to the calculations explained in Section III-C for the combinational locks [7]. This technique chooses the transistor sizes randomly and does not have a constraint on the number of keys that give the desired bias. Hence, the defense time depends only on the ratio of I_B to I_{ref}. This dependence enables the creation of a lock with a key size of 512 bits, as shown in Table III. As this technique can give the desired I_B for more than one key, the attack returns more than one correct key. However, the SMT solver is not called iteratively to find all possible

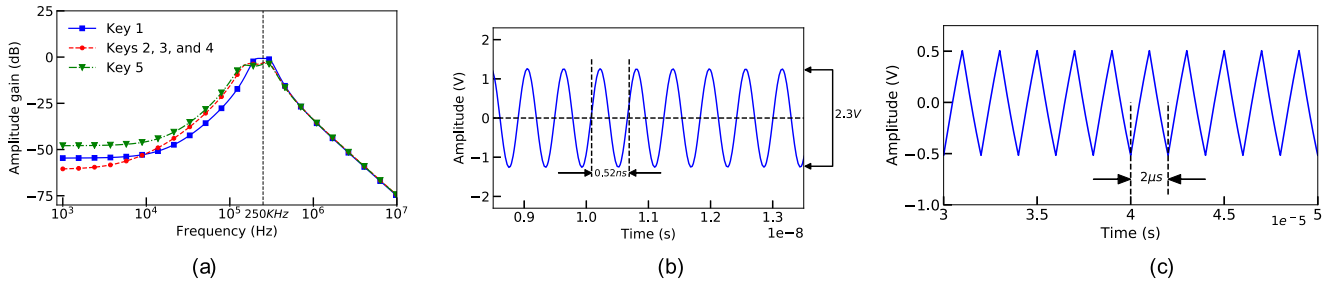

Fig. 10. (a) BPF simulations for the keys found by proposed attack on an 84-bit key combinational lock [7]. The attack returns five keys, in which key 1 is the correct key. The output response shows a 2 dB degradation in the gain at the ω_c for remaining keys. (b) Attack returns exactly one key for LC oscillator locked using combinational lock [7]. (c) Time period of oscillation for the unlocked TWG is 2 μs.

keys. The run is exited as soon as one of the valid keys is found.

E. Attack Results on Memristor-Based Protection [6]

The resistance value programmed into the memristor depends on the programming pulse voltage V_{PROG}, its duty cycle ρ, its frequency ω_{VT}, and the initial resistance value in the memristor. For a 2×2 crossbar, the designer has programmed the resistivity of all the memristors in the crossbars with 2 kΩ, the amplifier is designed with the gain A of 5, and V_{PP} is set to 1 V to produce $V_{\text{PROG}} = 2.5$ V. The attack returns the correct 8-bit key in 0.012 s. This experiment is repeated for increasing crossbar sizes and hence increasing key sizes. Fig. 7 shows that even for a 7×7 crossbar with a key size of 98 bits, the attack time is as low as 0.3 s.

F. Attack Results on Analog Performance Locking [10]

We executed the SMT-based attack on the ANN core of various sizes. The equation linking the I_B or V_B and the circuit specifications of the analog circuit protected such as gain is determined. For example, the equation linking the gain of BPF with the I_B is $G_{\text{BPF}} = (g_{m_1}C_1/g_{m_3}C_2)$ and $I_B = (g_m^2/\mu C_{\text{ox}}(W/L))$. Once the required I_B is calculated, it is given to the SMT solver along with the mathematical model of the ANN shared in Section II-F. The solver returns the input voltages and the weights of each synapse required to produce the necessary I_B or V_B. The defense time is the time required to determine the synapse weights, the neuron types, and the required input voltages for the given neural network size and the required I_B or V_B. Similarly, the attack time corresponds to the time taken to determine the synapse weights, the neuron types, and the required input voltages for the given neural network size and the required I_B or V_B range, as tabulated in Table V. The SMT solver is called once to determine the weights of the synapse, type of the neurons, and the input voltages to the neural network to provide the necessary I_B or V_B.

G. Attack Results on Camouflaged Analog IPs [9]

We demonstrate this attack on the fourth-order Gm-C BPF using TSMC 180-nm multi-V_{th} technology. The size ratios of each of the multi-V_{th} transistors with respect to NVT transistors are 1, 0.65, and 0.39, respectively. If the current mirror transistors are camouflaged, both the reference and the mirroring transistor have the same V_{th} as their gates are shorted. All the 80 transistors in the BPF are camouflaged. Transforming this netlist into a logic-locked netlist has 3^{80} key combinations with a key size of 160 bits. Apart from the 16 transistors which affect the circuit specification, there are 64 transistors in the current mirrors. The V_{th} type of the 16 transistors can be found by solving equations (A) and (C) listed in Table II for [9]. The V_{th} type of the current mirror transistors is not necessary to determine the V_{th} type of the other transistors, as the attacker can get the size ratio between the reference and mirroring transistors from the reverse-engineered netlist. The reference and mirroring transistors in the current mirrors are considered to be connected to the same key. Hence, there are effectively only 32 transistors. Each of the transistors can be one of the three variants. The total number of possible keys is 3^{32}. However, our attack returns only one correct key. Thus, our attack is also effective against current analog camouflaging [9].

H. Attack Results on AMSlock [11]

The attack is demonstrated on three analog circuits: a BPF, an LNA, and an LDO locked using [11]. The input size of the optimizer is n, the key size is k, and the HD is h. The effective key size is given by $k - \log_2 \binom{k}{h}$. This should be >80 for SAT attack resilience. As the locking of the digital optimizer is done manually in the RTL level, the defense time is not included in Table IV. The attack time required to break each of the setups is given in Table IV. It considers the time taken by the SAT solver to determine the correct key. As tabulated in Table IV, the SAT solver is called only once for all the circuits to determine the correct key to unlock the digital optimizer. For SFLL-HD^0, the number of PIPs is equal to one, and hence, PIP = key. Thus, (8) can be reduced to $\text{CNF}_{\text{locked}} \wedge \text{PIP} \wedge O$. The attack time for increasing key size is listed in Table IV.

In the case of SFLL-HD^h, the number of PIPs is given by $2^{n-k} \times \binom{k}{h}$. Also, for an n-bit input size, there are 2^n possible INs to the locked optimizer. However, as the analog circuit controls the input to the locked optimizer, there are only 1024 INs. Due to this constraint, the attacker knows only a subset of the PIPs, but not all. Hence, there can be more than one correct key, which gives the correct output for all the PIPs in the 1024 INs. As the recent attacks on SFLL-HD^h such as SFLL-hd–Unlocked [20] and FALL attack [21] requires all the PIPs in the attack formulation, we cannot reuse these attacks. Therefore, we use the attack formulations given in (8) to find the correct key. This correct key ensures the correct output for the 1024 INs. As the HD between the key and PIP increases from 0 to $(k/2)$, the number of PIPs protected by this key also increases (as indicated in [11, Fig. 6]). This means that for a given m number of PIPs, the number of keys that are at the same HD from these m PIPs will reduce. Hence, the time taken to find out these keys from 2^k possible keys

increases. To deduce the key using the attack formulations shared in Section II-H, an SMT solver or an SAT solver can be used. However, the increasing trend in the attack time is independent of the type of solver used and is rather dependent on the defense technique (SFLL-HD^h).

I. Attack Results on Shared Dependencies [14]

The time taken by the attack is the sum of the times required to unlock the analog and digital sections individually. The attack time to unlock the analog section is the time taken by the SMT solver to solve the attack equations and determine the key. The attack time to unlock the digital section is the time taken by the SAT solver to determine the correct key. As the analog section is locked using the parameter-biasing obfuscation technique, the SMT solver is called only once to determine one of the correct keys. Likewise, for the digital section locked using SFLL, the SAT solver is called once to determine the correct key, as indicated in Table V. The defense technique illustrated in Fig. 6 is implemented for different key sizes given in Table V. If the total key size is equal to 160 bits, 80 bits exclusively control the analog lock and 80 bits exclusively control the digital lock.

The attack formulations to find the key to unlock analog circuit is the same as the formulations given in Section III-D. As the digital circuit is locked using RLL and SFLL-HD^0, we assume the entire key to the digital circuit is bifurcated to RLL and SFLL, as shown in Table V. The circuit locked using RLL can be compromised using the SAT attack. The circuit locked using SFLL can be unlocked using the formulations shared in Section II-H.

IV. Discussion

A. What Is the Impact of the Reduced Current Range in Combinational Locks [7] on the Attack?

The I_B value is outside the range $(I_{B,lo}, I_{B,hi})$ for an incorrect key, where $I_{B,lo}$ is the minimum I_B and $I_{B,hi}$ is the maximum I_B. To ensure the attacker suffers from significant error (e.g., denial of service), the defender chooses the bias range widely. Therefore, the sets $I_{B,lo} = 0.2I_B$ and $I_{B,hi} = 4I_B$. The attacker finds the bias range using the circuit specification. However, this range is a subset of the bias range designed by the defender, e.g., $(0.9I_B, 1.1I_B)$. Hence, as long as this bias relationship is true, our attack can find the correct key.

We attack the locked circuits with the reduced range set by the defender to stress-test our proposed methodology. In Fig. 11, as the bias range reduces, the number of keys found by the attack increases. For the defense setup on BPF, where the defender's bias range is $(0.8I_B, 1.2I_B)$ and the range determined by the attacker is $(0.7I_B, 1.3I_B)$, the attack returns the keys whose corresponding bias values are in the range $(0.7I_B, 0.8I_B)$ and $(1.2I_B, 1.3I_B)$. For a key size of 56 bits, the search space is reduced from 2^{56} to 1190, as illustrated in Fig. 11. The attacker can now brute-force the determined keys to find the correct key using the oracle's response. This experiment is repeated for increasing ranges in the I_B determined by the attacker using the circuit specification. As the range increases, more number of keys are reported. Thus, based on the knowledge the attacker has on the analog circuit, he can determine the allowable I_B range and prune the unwanted keys.

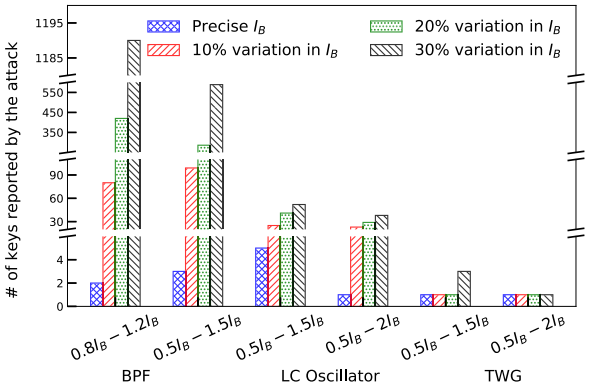

Fig. 11. Effect of reduced current range in the combinational locked circuits [7].

TABLE VI
Number of Keys Reported Based on the % Variation in the Transistor Dimensions

Circuit	Key size	Imprecise RE	
		2% variation	5% variation
BPF	84	18,866	20,008
LC osc.	14	100	637
Quad osc.	12	1	4
TWG	15	1	1

B. What Is the Impact of Approximate Models in Bias Calculations?

In this work, the quadratic expression models are used to estimate the I_B. Though these estimations have considerable inaccuracies compared to the charge-based model like an advanced compact model (ACM) and weak/moderate/strong inversion models of the transistors, it does not impact the attack results on the combinational lock [7]. This is because the defender's bias range is much larger (20%–400% of I_B) compared to the range estimated by the attacker. Also, in parameter-biasing obfuscation [8], memristor-based protection [6], shared dependencies [14], and analog performance locking [10], the attack returns the keys which produce any bias value within the bias range calculated by the attacker. As this range is calculated from the circuit specifications, the unlocked circuits' output response adheres to the circuit specification. Hence, our attack can determine the correct key even by using the approximate models.

C. What Is the Impact of Imprecise RE on the Attack?

RE is an expensive and error-prone process. This process is based on the etching each layer of the chip and taking high-resolution images of each layer. These images are annotated to get the reverse-engineered netlist. To the best of our knowledge, there are no published works that mention the RE process's inaccuracy. The standard RE process involves the same wet or plasma etching for delayering each metal layer [35] that is used in layout designing. Hence, we consider the minimum resolution in the layout designing as the impreciseness in the RE process. For the 180-nm technology node, the minimum resolution is 10nm. Hence, the variation in the transistors' physical dimensions due to errors in RE is considered equivalent to $\pm$ 5.56%. Hence, the impreciseness is modeled by giving error ranges of 2% and 5% to the transistors' physical dimensions. As the number of transistor

TABLE VII

ATTACK SUCCESS ON VARIOUS ANALOG LOCKS. ○ DENOTES LOCKED NETLIST. ● DENOTES THE ORACLE ACCESS. ⊥ DENOTES THAT THE RESOURCE AVAILABILITY AIDING OVERPRODUCTION BUT DOES NOT AID PIRACY. ø DENOTES REVERSE-ENGINEERED LOCKED NETLIST. ∗ DENOTES AVAILABILITY OF DIGITALLY CONTROLLED CURRENT SOURCE. ✓ DENOTES THAT THE DEFENSE IS BROKEN. ≈ DENOTES THE ATTACK REDUCE KEY SEARCH SPACE

Type	Defense	Circuit locked	TM (defense)	Claimed resilience	TM (attack)	SMT?	SAT?	Broken?
Analog-only locks	Combinational lock [7]	analog	○ ●	IP piracy	○ ●	✓	×	✓
	Parameter-biasing obfuscation [8]		○ ●	IP piracy	○ ● ⊥	✓	×	✓
	Memristor-based protection [6]		○ ●	IP piracy	○ ● ⊥	✓	×	✓
	Analog camouflaging [9]		ø ●	Reverse engineering	ø ●	✓	×	≈
	Analog neural network [10]		○ ●	Illegitimate access	○ ● ∗	✓	×	✓
AMS locks	AMS lock [11]	digital	○ ● ⊥	Overproduction	○ ● ⊥	×	✓	✓
	Shared dependencies [14]	analog and digital	○ ●	IP piracy	○ ● ⊥	✓	✓	✓

switches increases, the number of keys reported by our attack also increases, e.g., BPF, as shown in Table VI. Therefore, under the impact of RE, though our attack cannot find the correct key, it can considerably reduce search space.

D. What Is the Impact of PVT Variations on the Attack?

The minimum and maximum values of the circuit parameters, such as f_c, Q, and ω_{osc}, available in the circuit specification are considered in our attack algorithm, as given in (4). These values are calculated by the designer considering the impact of PVT variation. Hence, the attacker need not worry about the PVT impact on the attack equations. Note that the designer designs the circuit and the key such that the correct key should yield the desired performance, even in the presence of variations. Otherwise, he/she cannot sell on the market. Therefore, we also assume that variations in the output responses of the oracle are not possible beyond the ranges quoted in the specification. Therefore, the key found based on the bias range calculated using these specifications by the attacker should give the desired circuit performance. The correctness of this key is verified by comparing the unlocked circuit's output response with the oracle's response.

E. Can the Attacker Design the Circuit From Scratch Instead of Attacking?

This design process is cumbersome and time-consuming. When the attacker knows only the expected performance and the specifications, it requires a process with many iterations to build a functional chip. Given that the design process heavily depends on expertise, skills, and even luck (the number of iterations required to get the desired performance), it is difficult to estimate the design time. Thus, it is easier for an attacker to "steal" an existing design, which is what many of the existing defense techniques aim to prevent.

F. Can This Attack Help an Attacker Who Can Pirate or Overproduce the Chip to Find the Correct Key?

In the case of IP piracy, assume the attacker obtains the precise value of the bias using (3). The locked bias circuit can be replaced with a circuit generating this required bias input. This replacement makes the analog circuit functional. However, if he obtains only the range of the bias input using (4), he cannot replace the locked circuit as he does not know the precise I_B value. Hence, the SMT solver is fed with (1), (2), and (4) to return the correct key and the correct bias value. The attacker can then manufacture the chip replacing the CCM with the current mirror that generates the required I_B, thereby enabling him to pirate the chip. Now, let us consider overproduction. Here, the attacker can only overproduce the chip, but netlist level modifications are not feasible. He either obtains the precise value or the range of the bias. The corresponding equations are solved using SMT solver to find the correct key. Hence, our technique unlocks the chip even if the attacker has little control in the foundry.

G. What Is the Implication of This Attack on Various Analog Locks?

None of the defenses are resilient against IP piracy, as shown in Table VII. In the combinational lock, each chip has a unique user key. This key is XORed with the input from the chip identification unit, such as physically unclonable functions, to produce the common key. This common key controls the CCM [7]. The SMT formulations in Section II-D helps in determining the common key using the locked netlist and circuit specification. However, to remove the dependence on the unique key, the attacker has to remove the chip identification unit from the layout. Hence, the attacker should have the necessary resources to modify the layout. Unlike [7], the parameter-biasing obfuscation [8] and memristor-based protection [6] do not have two sets of keys. The SMT formulations in Section II-D and Section II-E help in determining the key for [8] and [6], respectively. The resources required to overproduce the chip are sufficient to break this technique.

In analog camouflaging [9], the attacker does not have access to the foundry, and hence, our SMT formulations can only reduce the search space. The analog performance locking [10] can be broken only if the attacker can have access to a standalone DCCS, which is integrated into the ANN. This helps in programming the precise weights in the synapses. Our SAT formulations in Section II-H can determine the correct key using the overproduction threat model, thus compromising the AMSlock [11]. This defense claims resilience against overproduction. In shared dependencies [14], the SAT formulations are used to determine the correct key to unlock the digital part, and SMT formulations are used to unlock the analog part. Though the attack claims resilience against IP piracy, it could be broken without any manual modification in the layout (mask) of the locked chip.

V. CONCLUSION

Our SMT attack has shown the vulnerabilities in the existing analog-only locking techniques [6]–[10], thereby enabling the

attacker to find the key to pirate, overproduce, or reverse engineer a chip. The attack time does not depend on the key size; we demonstrated our attack on a 512-bit key. Additionally, we extended this attack on camouflaged analog IPs [9], where we could reduce the key search space from 3^{80} to 3^{32}, thereby enabling the brute-force attack to find the correct key. We can also successfully break AMS locking techniques [11], [12], [14] by providing the required SAT formulations to compromise the SFLL-HD^h lock used in securing the digital part of the AMS circuits. Analog security, being a fast-growing field, has two new defenses, [36] and [37]. The former proposes a technique to secure programmable analog ICs, while the latter prevents unauthorized access to analog ICs using floating gate transistors. We shall evaluate the resilience of these techniques in our future works. Also, while digital locking techniques have now obtained reasonable solutions against oracle-based attacks, we urge the community to undertake a theoretical approach in developing defenses for analog circuits.

ACKNOWLEDGMENT

The authors thank Qualcomm and Synopsys for their support in this work. They would also like to thank Jiafan Wang for his help.

REFERENCES

[1] Defense Science Board. (2005). *Defense Science Board (DSB) Task Force on High Performance Microchip Supply*. Accessed: Jan. 6, 2020. [Online]. Available: https://dsb.cto.mil/reports/2000s/ADA435563.pdf

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

[3] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. 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.

[4] IHS Technology Press Release. (2012). *Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market*. Accessed: Jan. 6, 2020. [Online]. Available: http://technology.ihs.com/405654/top-5-most-counterfeited-parts-represent-a-169-billion-potential-challenge-for-global-semiconductor-market

[5] A. Hastings, *The Art of Analog Layout*. London, U.K.: Pearson, 2005.

[6] 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.

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

[8] V. V. Rao and I. Savidis, "Parameter biasing obfuscation for analog IP protection," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2017, pp. 1–6.

[9] A. Ash-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.

[10] G. Volanis, Y. Lu, S. G. R. Nimmalapudi, A. Antonopoulos, A. Marshall, and Y. Makris, "Analog performance locking through neural network-based biasing," in *Proc. IEEE 37th VLSI Test Symp. (VTS)*, Apr. 2019, pp. 1–6.

[11] 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*, Nov. 2018, pp. 7:1–7:8.

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

[13] 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 *Proc. 16th Int. Conf. Synth., Modeling, Anal. Simulation Methods Appl. Circuit Design (SMACD)*, Jul. 2019, pp. 185–188.

[14] K. Juretus, V. V. Rao, and I. Savidis, "Securing analog mixed-signal integrated circuits through shared dependencies," in *Proc. Great Lakes Symp. VLSI*, May 2019, pp. 483–488.

[15] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2015, pp. 137–143.

[16] K. Z. Azar, H. M. Kamali, H. Homayoun, and A. Sasan, "SMT attack: Next generation attack on obfuscated circuits with capabilities and performance beyond the SAT attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 1, pp. 97–122, Nov. 2018.

[17] Y. Xie and A. Srivastava, "Delay locking: Security enhancement of logic locking against IC counterfeiting and overproduction," in *Proc. 54th Annu. Design Automat. Conf.*, Jun. 2017, pp. 1–6.

[18] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal attacks on logic locking and camouflaging techniques," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 517–532, Apr. 2020.

[19] X. Xu, B. Shakya, M. M. Tehranipoor, and D. Forte, "Novel bypass attack and BDD-based tradeoff analysis against all known logic locking attacks," in *Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 189–210.

[20] F. Yang, M. Tang, and O. Sinanoglu, "Stripped functionality logic locking with Hamming distance-based restore unit (SFLL-HD)—Unlocked," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2778–2786, Oct. 2019.

[21] D. Sirone and P. Subramanyan, "Functional analysis attacks on logic locking," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 936–939.

[22] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proc. 48th Design Automat. Conf. (DAC)*, 2011, pp. 333–338.

[23] C.-S. Chang, C.-P. Chao, J. G. I. Chern, and J. Y.-C. Sun, "Advanced CMOS technology portfolio for RF IC applications," *IEEE Trans. Electron Devices*, vol. 52, no. 7, pp. 1324–1334, Jul. 2005.

[24] Texas Instruments. (2010). *Universal Active Filter*. Accessed: Apr. 25, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/uaf42.pdf

[25] Chipworks. (2016). *Reverse Engineering Software*. Accessed: Apr. 29, 2020. [Online]. Available: https://www.techinsights.com/

[26] T. Iizuka, *CMOS Technology Scaling and Its Implications*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[27] R. Sotner, J. Jerabek, N. Herencsar, K. Vrba, and T. Dostal, "Features of multi-loop structures with OTAs and adjustable current amplifier for second-order multiphase/quadrature oscillators," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 5, pp. 814–822, May 2015.

[28] Z. Zahir and G. Banerjee, "A multi-tap inductor based 2.0–4.1 GHz wideband LC-oscillator," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2016, pp. 330–333.

[29] R. Senani, D. R. Bhaskar, V. K. Singh, and R. K. Sharma, *Sinusoidal Oscillators and Waveform Generators Using Modern Electronic Circuit Building Blocks*. Cham, Switzerland: Springer, 2015.

[30] Texas Instruments. (2012). *AWR1243 Single-Chip 77-GHz and 79-GHz FMCW Transceiver*. Accessed: Jan. 6, 2020. [Online]. Available: http://www.ti.com/lit/ds/symlink/awr1243.pdf

[31] J. P. Lang. (2017). *iSAT3*. Accessed: Jan. 6, 2020. [Online]. Available: https://projects.informatik.uni-freiburg.de/projects/isat3/wiki/ISAT3_004

[32] M. Yasin and O. Sinanoglu, "Transforming between logic locking and IC camouflaging," in *Proc. 10th Int. Design Test Symp. (IDT)*, Dec. 2015, pp. 1–4.

[33] S. Lee *et al.*, "A built-in self-test and *in situ* analog circuit optimization platform," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 10, pp. 3445–3458, Oct. 2018.

[34] I. Schnell. (2013). *Pycosat 0.6.3*. Accessed: Jan. 6, 2020. [Online]. Available: https://pypi.org/project/pycosat/

[35] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Cryptographic Hardware and Embedded Systems*. Springer, 2009, pp. 363–381.

[36] M. Elshamy, A. Sayed, M.-M. Louerat, A. Rhouni, H. Aboushady, and H.-G. Stratigopoulos, "Securing programmable analog ICs against piracy," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 1–7.

[37] G. Volanis, A. Antonopoulos, S. G. R. Nimmalapudi, A. Marshall, Y. Makris, and Y. Lu, "Range-controlled floating-gate transistors: A unified solution for unlocking and calibrating analog ICs," in *Proc. Design, Automat. Test Eur. Conf. Exhib.*, 2020, pp. 286–289.