MixLock: Securing Mixed-Signal Circuits via Logic Locking

Julian Leonhard, Muhammad Yasin, Shadi Turk, Mohammed Thari Nabeel, Marie-Minerve Louërat, Roselyne Chotin-Avot, Hassan Aboushady, Ozgur Sinanoglu, Haralampos-G. Stratigopoulos

MixLock: Securing Mixed-Signal Circuits via Logic Locking

Julian Leonhard*, Muhammad Yasin†, Shadi Turk‡, Mohammed Thari Nabeel§, Marie-Minerve Louërat*, Roselyne Chotin-Avot*, Hassan Aboushady*, Ozgur Sinanoglu§, Haralampos-G. Stratigopoulos*
*Sorbonne Université, CNRS, LIP6, Paris, France
†New York University, New York, USA
‡Seamless Waves, Paris, France
§New York University Abu Dhabi, Abu Dhabi, UAE

***Abstract*—In this paper, we propose a hardware security methodology for mixed-signal Integrated Circuits (ICs). The proposed methodology can be used as a countermeasure for IC piracy, including counterfeiting and reverse engineering. It relies on logic locking of the digital section of the mixed-signal IC, such that unless the correct key is provided, the mixed-signal performance will be pushed outside of the acceptable specification range. We employ a state-of-the-art logic locking technique, called Stripped Functionality Logic Locking (SFLL). We show that strong security levels are achieved in both mixed-signal and digital domains. In addition, the proposed methodology presents several appealing properties. It is non-intrusive for the analog section, it incurs reasonable area and power overhead, it can be fully automated, and it is virtually applicable to a wide range of mixed-signal ICs. We demonstrate it on a $\Sigma\Delta$ Analog-to-Digital Converter (ADC).**

I. Introduction

An Integrated Circuit (IC) may be subject to various types of attacks during its lifetime. These attacks are launched by a knowledgeable adversary and may have different incentives. Threats can be classified into four main categories, namely IC piracy, which includes reverse engineering and counterfeiting, hardware Trojans, side-channel attacks, and fault injection attacks.

The term reverse engineering refers to the derivation of IC proprietary information, i.e. architecture, netlist, layout, etc. It aims at reducing the attacker's technological disadvantage against the "author" of the IC, gathering necessary information for producing a similar or identical IC, e.g. a counterfeit, or locating the root-of-trust part of the IC and stealing secret information, such as cipher keys. The term counterfeit refers to (i) an illegally cloned IC that is sold as original, (ii) a used and possibly aged IC that is illegally recycled and resold as new, or (iii) ICs that are overproduced by an untrusted foundry and are illegitimately sold in after market. Hardware Trojans are malicious modifications in an IC aiming at covertly leaking secret information, degrading the reliability and performance, or rendering the IC completely malfunctional. Side-channel attacks are based on non-invasive observation of electrical or physical characteristics of the IC through the design-for-test infrastructure, power analysis, electro-magnetic analysis, etc., and aim at leaking secret information. Fault injection attacks consist of triggering an event in the IC, such as voltage glitches in the power supply, which may have a devastating effect on the functionality of the IC, may reduce reliability and performance, or may be used for leaking secret information.

Hardware security refers to understanding security breaches and developing countermeasures for resiliency against the aforementioned threats. Addressing these threats is considered of major significance, especially in the case of ICs deployed in sensitive sectors, such as defense, infrastructure, health, automotive, space, and telecommunication applications.

While extensive research efforts have been expended over the last decade in understanding trust and security threat scenarios in digital ICs and developing solutions [1]–[5], the topic remains largely unexplored for analog ICs and there is an alarming lack of understanding of the solution space [6]–[8]. Analog ICs are perhaps the weakest link in warranting the global security policy for the entire electronic system.

This work deals with hardware trust and security aspects specifically for mixed-signal ICs, which is a large subclass of analog ICs, including data converters, Phase Locked Loops (PLLs), Radio Frequency (RF) transceivers, etc. In particular, we develop a locking methodology for mixed-signal ICs, called *MixLock*, that prevents IC piracy. Locking aims at transforming the original design into one that is functionally equivalent, but requiring a secret key to unlock the correct functionality. Applying an invalid key will result in dramatically degraded mixed-signal performance. In *MixLock*, locking is achieved via logic locking (aka logic encryption) of the digital section of the mixed-signal IC. For this purpose, we employ a state-of-the-art logic locking technique, called Stripped-Functionality Logic Locking (SFLL) [9]. Metrics are proposed to quantify the analog security level, i.e. the mixed-signal functionality corruption for invalid keys. The digital security level is expressed in terms of resilience to all known logic locking attacks. We show that *MixLock* is capable of co-optimizing security in the analog and digital domains. In addition, *MixLock* presents several appealing properties. It is non-intrusive for the analog section, it incurs reasonable area and power overhead, it can be fully automated, and it is virtually applicable to a wide range of mixed-signal ICs. We demonstrate *MixLock* on a $\Sigma\Delta$ Analog-to-Digital Converter (ADC).

The rest of the paper is structured as follows. In Section II, we discuss previous work on analog IC locking. In Section III, we provide an overview of *MixLock*. In Sections IV and V, we discuss the threat model and the attack resilience analysis. In Section VI, we discuss logic locking. In Section VII, we present the case study. In Section VIII, we present our experimental results. Section IX concludes the paper.

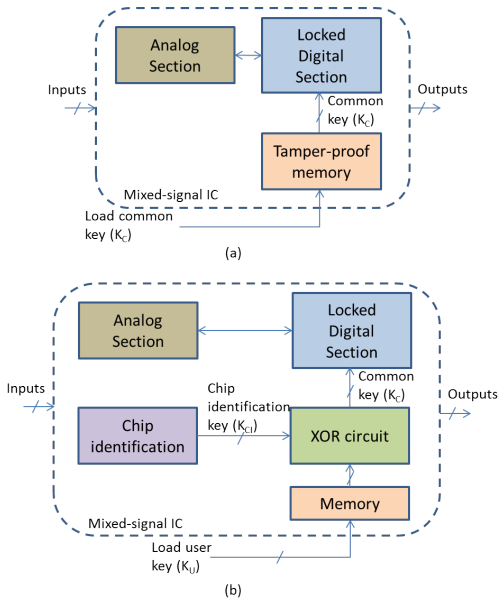

Fig. 1. Mixed-signal IC locked with *MixLock*.

II. Previous Work on Analog IC Locking

In [10], the sense amplifiers used to read out the contents of a memory are locked via the body biasing of one of the transistors in their input differential pair. The locking scheme is based on memristors. The usage of memristors in this context is interesting; however, fabricating memristors requires a special process and still remains as an "exotic" technology.

In [11], it is proposed to replace transistors within the biasing circuit with parallel-connected transistors whose gates are controlled by key bits. The key bits enable transistors whose aggregate width equals that of the original transistor.

In [12], it is shown how to redesign the current mirrors providing the biasing so as to insert key-bits. Extra branches are inserted, where each branch is comprised of the mirroring transistor and possibly several switches that are controlled by the key-bits. The resultant biasing circuit will depend on which branches are switched-on, as well as on the geometry of the mirroring transistor in these branches. The proposed approach guarantees that only one key unlocks the functionality.

All three aforementioned approaches [10]–[12] are vulnerable to removal attacks since a smart attacker can simply remove the obfuscated biasing circuit and replace it with a "fresh" one with no locking mechanism. The attacker does not have to recover the key; it suffices to recover the biases, which typically are not so many.

III. Proposed Technique: MixLock

MixLock aims at locking a mixed-signal IC via a logic locking mechanism embedded into its digital section, as illustrated in Fig. 1. Only when the valid secret key is provided, referred to as common key K_C, the correct functionality is unlocked, that is, the digital section implements the correct function for any input. Logic locking will be discussed in more detail in Section VI.

The common key K_C can be stored directly in a tamper-proof memory, as shown in Fig. 1(a) [13]. Alternatively, the locking system in Fig. 1(b) can be used [14], which employs a chip identification key K_{CI} that is unique for every IC and can be generated, for example, by a Physical Unclonable Function (PUF) [15]. The unique user key K_U for the IC that is finally given to the user is the one that, when XORed with the chip identification key K_{CI}, generates the common key K_C. The common key K_C and chip identification key K_{CI} are kept secret and should not be shared with an untrusted party.

The digital section is typically part of a signal-processing chain or part of a feedback loop and, according to the mixed-signal IC type, can perform different whole functions or sub-functions. The underlying idea is that logic locking of the digital section becomes a means for corrupting the mixed-signal IC performance trade-off. The objective is that unless the valid key is provided, the performance trade-off is locked, that is, one or more performances lie outside their acceptable specification range.

MixLock presents several appealing properties:

Non-intrusive. It is non-intrusive since it does not alter the analog section and since any performance degradation in the digital section can be easily absorbed with no degradation in the mixed-signal performance. This is key for its wide adoption by analog designers.

Low-overhead. Typically die area and power consumption in a mixed-signal IC is largely dominated by the analog section. The area and power overhead in the digital section introduced by logic locking is already affordable considering the digital section alone; this overhead, when projected for the entire mixed-signal IC, will be even easier to justify.

Fully automated. Typically, design-for-X (DfX) techniques for mixed-signal ICs, where "X" can be test, reliability, calibration, diagnosis, etc., require significant extra design effort. DfX also needs to be revisited for every new product or new technology node. In contrast, the proposed Design-for-Trust (DfTr) *MixLock* technique is fully automated since it is based on logic locking of the digital section, which is fully automated.

Wide applicability. It can be virtually applied to a wide range of mixed-signal ICs that have a large digital section, including data converters, PLLs, RF transceivers, etc. It also fits well the general trend towards digitally-assisted analog designs and digital centric mixed-signal architectures, where the goal is to make a thoughtful shift of functionality from the analog into the digital domain, in order to alleviate analog design complexity and enable post-manufacturing tuning, self-calibration, and reconfigurability.

IV. Threat Model

MixLock is intended to serve as a countermeasure against mixed-signal IC piracy, which can be broken down into several distinct threats. These threats, the assumptions on the capabilities of the adversary, and the conditions under which *MixLock* delivers resilience are described next.

Reverse engineering. We assume that the adversary has full capabilities to extract the architecture, netlist, layout, etc. Even in this case, the adversary will not be able to reveal the exact functionality as the common key is unknown. Of course, a smart adversary can quickly realize by tracing the key bits structurally that the digital section is locked. In this case, the

digital section can be removed and replaced with a "fresh" one with no locking mechanism. But this requires that the adversary has the required design expertise and is willing to spend some significant design effort, given also that the design of the digital section is heavily tightened to that of the analog section. The requirement for significant redesign effort clearly goes against the original incentive of the adversary; thus, *MixLock* provides good resilience against this threat.

Cloned counterfeits. The common key is unknown; thus, with *MixLock* in place, a cloned counterfeit is practically unusable.

Recycled counterfeits. *MixLock* does not provide any protection, unless the scheme in Fig. 1(b) is used and the user key is reloaded every time the IC is powered up. But arguably there are simple techniques to detect recycled ICs, for example, through the use of on-chip lightweight sensors [16].

Overproduced counterfeits. *MixLock* provides protection as long as one of the following approaches is used: (a) The test is performed in a trusted test facility; (b) The ICs after fabrication are sent from the fab to the trusted party, i.e., the design house, that loads the common key using the scheme in Fig. 1(a) and sends them back to the untrusted facility for testing; (c) The trusted party remotely activates the chip for testing using asymmetric cryptography [17].

V. Security Analysis

The security level of *MixLock* is defined in terms of security level of the underlying logic locking, called *digital security level*, and the security level of performance trade-off locking, called *analog security level*. An attacker can try to unlock explicitly the digital section without caring about the analog section, or can try to unlock directly the performance trade-off, i.e., achieve a satisfactory performance trade-off, which perhaps can be achieved with an incorrect common key.

The digital security level is measured by the effort that the designer must spend for identifying the common key. It is dictated by the resilience against known attacks, as will be explained in more detail in Section VI.

We propose to measure the analog security level using different metrics in the analog domain, namely, *error rate*, *mean absolute error*, and *minimum error*. *Error rate* is the percentage of incorrect keys resulting in violation of one or more performance specifications. The *error rate* may be misleading if for incorrect keys the violated performance(s) is (are) slightly outside the(ir) specification(s). To account for this scenario, we also use the *mean absolute error* metric defined as the average absolute performance difference between the unlocked mixed-signal IC and locked versions. *Minimum error* is the minimum observed performance difference between the unlocked mixed-signal IC and locked versions, indicating the *worst-case* locking. These metrics can be quantified by putting to a test a large set of random incorrect keys.

Attacks to unlock directly the performance trade-off are not known at this point. A possible scenario is that the attacker uses optimization algorithms, such as gradient descent, simulated annealing, etc., to search for a common key that brings the performances within the acceptable specification range. Such an attack is very unlikely to succeed since the mixed-signal performances are not related to the key through a well-behaved and smooth function. The optimization is likely to "zigzag" endlessly. Especially if the key width is large and if the *minimum error* defined above is large, this attack is doomed to fail.

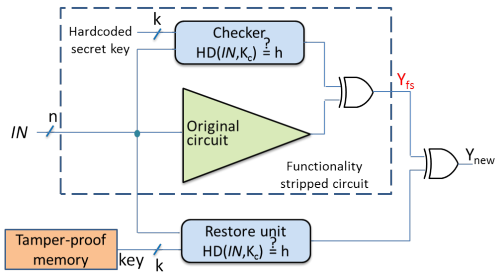

Fig. 2. SFLL architecture.

VI. Logic Locking

Logic locking protects the digital circuit by modifying it and adding new logic into it, such that its functionality is controlled by a key. The earliest traditional logic locking techniques, e.g. Random Logic Locking (RLL) [17], Fault analysis-based Logic Locking (FLL) [18], and Strong Logic Locking (SLL) [19], aimed at inserting key gates into the design that are controlled by key-bits, which compose the key. The best key gate locations are determined while balancing the security objectives and the implementation overhead. However, the SAT attack [20], which is based on a Boolean satisfiability solver, was able to break all these techniques and recover the secret key with very reasonable effort. Techniques to thwart SAT attack, e.g., SARLock [21] and Anti-SAT [22], were shown to be susceptible to removal attacks [23], which aim to identify and isolate the protection logic. These SAT-resilient techniques can be combined with traditional logic locking for improving the output corruptibility [24], yet such integration can be circumvented using the approximate attacks, e.g., AppSAT [25] and Double-DIP [26], which reduce the security level down to the one provided by the SAT-resilient technique and extract a key that establishes an incorrect but approximate functionality.

The state-of-the-art technique is SFLL [9] that achieves holistic security against SAT, removal, and approximate attacks in a quantifiable manner. As illustrated in Fig. 2, in the SFLL architecture, a part of the original circuit functionality is stripped away using a *checker*. In particular, for all input patterns that are Hamming distance h away from the secret key, the output of the original circuit is flipped. Only upon supplying the valid secret key, the *restore unit* cancels the errors introduced by the *functionality-stripped circuit*, recovering the original output.

Let k be the number of key-bits composing the key and let n be the number of inputs of the digital circuit. It can be shown that SFLL is $slSAT$-secure against SAT attack, where $slSAT = k - \lceil \log_2 \binom{k}{h} \rceil$, meaning that the SAT attack effort required to extract the secret key is equivalent to breaking a $k - \lceil \log_2 \binom{k}{h} \rceil$-bit key in a brute-force way [9]. It can also be shown that SFLL is $slREM$-resilient against removal attacks,

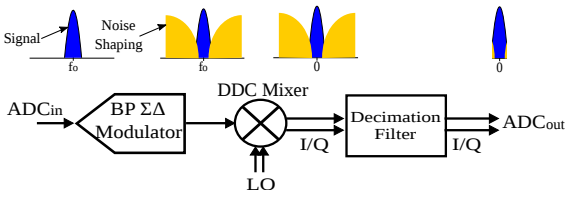

Fig. 3. The Bandpass ΣΔ ADC used as case study.

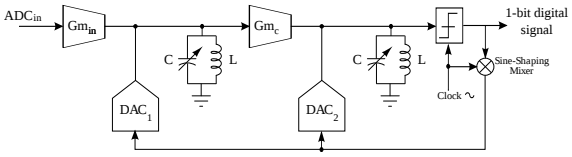

Fig. 4. The analog 4th order LC bandpass ΣΔ modulator.

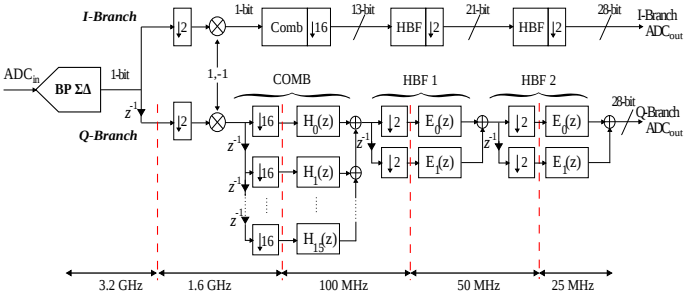

Fig. 5. The digital DDC mixer and decimation filter.

where $slREM = \binom{k}{h} \cdot 2^{n-k}$ is the number of protected input patterns, meaning that the larger the number of protected input patterns, the more intricate the changes to the original logic are, and, thereby, the harder it is for the removal attack to succeed [9]. Finally, it can be shown that SFLL is $slAPX$-resilient against approximate attacks, where $slAPX = \frac{\binom{k}{h}}{2^k}$ is the error rate [9], meaning that the higher the error rate, the more difficult it is to find a key that establishes approximate functionality. Therefore, SFLL allows a designer to trade-off the desired security level against different attacks by choosing appropriately k and h [9].

VII. Case Study: ΣΔ ADC

To demonstrate *MixLock*, we used as case study a bandpass (BP) ΣΔ ADC which converts a band $B = 25\,\text{MHz}$ centered at $F_0 = 2.4\,\text{GHz}$ with a sampling frequency $F_s = 3.2\,\text{GHz}$ [27]. The block-level schematic is shown in Fig. 3.

The analog section of the ΣΔ ADC is a 4th order LC BP ΣΔ modulator, shown in Fig. 4. The ΣΔ modulator converts the analog input signal to an oversampled low-resolution 1-bit digital signal with frequency F_s.

The digital section of the ΣΔ ADC, shown in Fig. 5, is composed of a Digital Down-Conversion (DDC) mixer and a multi-stage multi-rate decimation filter. The decimation filter removes the out-of-band noise from the ΣΔ modulator output and down-samples it, with a factor OSR=$\frac{F_s}{2B} = 64$, to convert it to a high-resolution 28-bit digital signal sampled at the Nyquist rate. The decimation filter is composed of a comb filter $COMB$, a first half-band filter $HBF1$, and a second half-band filter $HBF2$, with down-sampling factors of 16, 2, and 2, respectively. The DDC mixer offers an additional down-sampling factor of 2, thus the total down-sampling factor is 128.

The aim is to lock the SNR performance of the ΣΔ ADC via locking its digital section. Any incorrect key should result in an SNR performance that violates the specification, rendering the ΣΔ ADC unusable. The ΣΔ ADC has a nominal SNR of 70 dB with a specification set at 65 dB.

VIII. Results

A. Setup

A pure sinusoidal signal with $F_{in} = F_0 + \Delta F$, where $\Delta F = 20\,\text{KHz}$, is applied at the input of the ΣΔ ADC and 2^{20} samples are recorded at the output of the ΣΔ modulator corresponding to 13 input signal periods. This recorded output bitstream is used as input to the digital section.

We study several logic locking techniques in terms of their impact on digital and analog security. The digital section is transformed into the locked digital section at RT-level VHDL. The locked digital section is synthesized using the Encounter RTL Compiler with a $65nm$ CMOS low-threshold voltage library and appropriate timing constraints, in order to compute estimated area, power consumption, and performance overhead compared to the original version.

Evaluating the SNR for a given key involves loading the key into the digital section, simulating the digital section at RT-level VHDL using the recorded ΣΔ modulator output bitstream as input, and performing a Fast Fourier Transform (FFT) at the output of the decimation filter to calculate the SNR.

The analog security level metrics defined in Section V are calculated based on 10^3 randomly generated incorrect keys. This calculation is fully automated and, for any of the logic locking mechanisms that we considered, takes up around 40 minutes on a Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz with 8 GB of RAM.

B. Security analysis

Logic locking attacks, in general, assume the existence of scan chains. On the other hand, mixed-signal ICs do not always include scan chains into their digital section. When the amount of logic is large enough, using scan chains is a recommended test practice so as to increase defect coverage and diagnosability. However, even then, scan chains are not always used for various reasons. For example, the mixed-signal IC will be, after all, tested as a whole; inserting scan chains affects speed; many analog designers are not familiar with scan chains, etc. If scan chains are absent, then any logic locking technique can be used since logic locking attacks are inapplicable without scan access. Thus, it suffices to select the smallest-overhead technique that achieves strong analog security. If scan chains are present, then achieving strong digital security becomes another dimension of the problem.

We first implement the SFLL technique [9], using a secret key of k=128 bits. With the SFLL technique, we lock the Most Significant Bit (MSB) of the $COMB$ filter's output. In this context, locking a bit line means that we strip the functionality of the sub-circuit that drives the bit line. We chose this bit

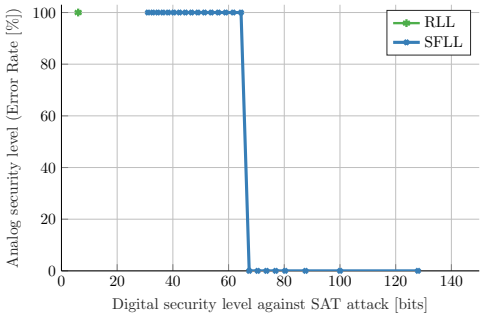

Fig. 6. Trade-off between analog security level in terms of *error rate* and digital security level against SAT attack for different logic locking techniques.

line aiming to introduce high functionality corruption early in the digital signal processing chain. We chose $h = 15$ since this choice leads to a strong 64-bit resilience against the SAT attack, and also strong resilience against removal and approximate attacks, according to the formulas in Section VI.

For this choice of SFLL parameters, we observed that not all secret keys result in functionality corruption that is high enough to achieve 100% error rate. This can be explained by the fact that SFLL protects a subset of input patterns, as discussed in Section VI. For a sinusoidal input signal, such as the one specified in Section VIII-A, the number of input patterns generated at the input of the functionality-stripped sub-circuit is limited and, thereby, for a random key it is likely that not enough of these input patterns are protected to achieve 100% error rate. To this end, we crafted a secret key to achieve 100% error rate for the selected sinusoidal input signal. The number of keys that meet this objective is very high and, in any case, the selected sinusoidal input signal based on which the key is crafted is unknown to the attacker. Note also that in a real application, inputs are not well-structured and well-behaved like a sinusoidal. For example, $\Sigma\Delta$ ADCs are the most popular choice for a variety of precision measurement applications and for voiceband and audio applications. Real-application signals are time-varying in nature, their spectral contents vary with time, they are rich in frequencies, etc. For these high-activity signals, the number of input patterns generated at the input of the functionality-stripped sub-circuit will be large, and, thereby, intuitively, any key will result in recurrent functionality corruption. Note that in [9], SFLL with $h = 0$ was used to lock a microcontroller designed using the ARM Cortex-M0 microprocessor. Choosing $h = 0$ implies only one protected input pattern, but still this was enough to break functionality.

We also implement the basic RLL technique [17] using the same secret key and locking the same sub-circuit. We observed that RLL achieves 100% error rate regardless of the secret key that is chosen.

Fig. 6 shows the trade-off between the analog security level, defined using the *error rate* metric, and the digital security level against the most powerful and lethal SAT attack. As it can be seen, RLL results in *error rate* of 100%, but it offers no resilience against the SAT attack in the case where scan chains are present. For the SFLL technique, the different points on

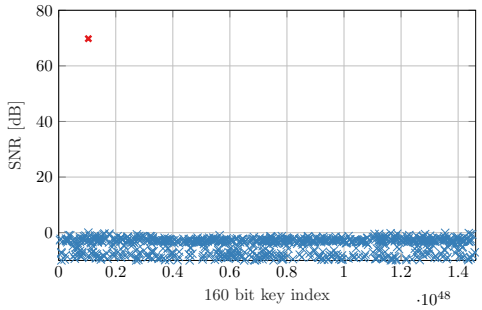

Fig. 7. SNR for 10^3 incorrect keys and the correct key.

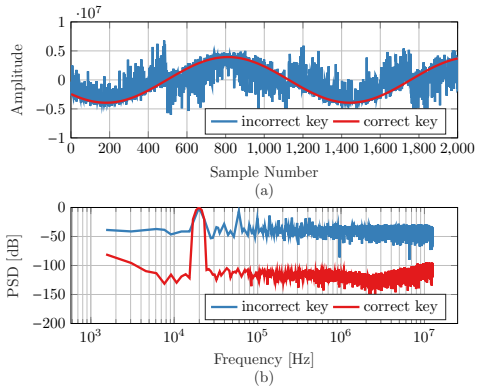

Fig. 8. Transient and frequency responses of the unlocked and a locked $\Sigma\Delta$ ADC.

the curve are produced by varying h. For $h = 15$, the sweet trade-off point 100% error rate and 64-bit resilience against the SAT attack is obtained. Regarding the other metrics for analog security, we obtain 54.5 dB *mean absolute error* and 54.5 dB *minimum error*. Choosing $h > 15$ will increase functionality corruption and, thereby, will improve further the *mean absolute error* and *minimum error* metrics, at the expense of decreased resilience against the SAT attack.

To increase functionality corruption, we can alternatively implement a compound technique. For example, a first SFLL mechanism with $k = 128$ and $h = 15$ can be intertwined with a second SFLL mechanism with $k = 32$ and $h = 16$, which locks the MSB-1 bit of the $COMB$ filter's output. We refer to this SFLL version as 1.5xSFLL. In fact, SFLL can be combined with other techniques too; for example, we can intertwine SFLL with RLL with $k = 32$. In theory, these compound techniques can be reduced to the first SFLL mechanism by applying AppSAT [25] or Double-DIP [26], as mentioned in Section VI, so they are appropriate only for the naive attacker.

Fig. 7 plots the SNR for 10^3 incorrect keys and the correct key using the 1.5xSFLL technique. The unlocked $\Sigma\Delta$ ADC stands out with a correct SNR of 70 dB. Locked versions have an SNR below the specification of 65 dB. Besides the 100% *error rate*, locking results in 70.6 dB *mean absolute error* and 65 dB *minimum error*. In fact, unless the correct key is provided, the input signal gets completely buried under the noise floor. Fig. 8 considers an arbitrarily selected incorrect key and compares the transient and frequency responses of the unlocked and a locked $\Sigma\Delta$ ADC. The locked $\Sigma\Delta$ ADC presents a large amount of glitches in its transient response,

TABLE I
OVERHEAD USING DIFFERENT UNDERLYING LOGIC LOCKING TECHNIQUES.

Technique	SFLL	RLL	1.5xSFLL	SFLL+RLL
Digital Section (%)				
Area	20.1	5.6	24.4	21.1
Power	29.5	9.3	35.3	30.9
Delay	19.5	3.76	22.2	21.8
$\Sigma\Delta$ ADC (%)				
Area	6.7	1.9	8.1	7.0
Power	9.8	3.1	11.8	10.3
Performance	0	0	0	0

which translate to a high noise floor in the frequency response, resulting in corrupted SNR.

C. Implementation Cost

Table I shows the overhead using the different underlying logic locking techniques SFLL, RLL, 1.5xSFLL, and SFLL+RLL. *MixLock* incurs overhead only for the digital section. This overhead is projected to the entire $\Sigma\Delta$ ADC considering that the digital section occupies about 30% of the die area and is responsible for about 30% of the total power consumption. The slack reserve in the critical path of the digital section is large enough to accommodate more gates; thus, the delay penalty gets easily absorbed and does not translate to an SNR performance penalty. Regardless of the employed logic locking technique, the unlocked $\Sigma\Delta$ ADC has an SNR of 70 dB, that is, there is no performance degradation due to locking. If scan chains are absent, then the basic RLL technique can be used since it provides lower overhead compared to SFLL. If scan chains are present, then SFLL achieves optimal all-around analog and digital security levels with an area and power overhead of 6.7% and 9.8%, respectively, which are very reasonable. For higher functionality corruption, one can use 1.5xSFLL or SFLL+RLL at the expense of slightly higher area and power overhead.

IX. CONCLUSION

Hardware security vulnerabilities have been addressed through various methods in the digital domain while similar solutions are largely missing in the analog domain. We proposed *MixLock* which protects mixed-signal ICs via locking their digital part. We developed security metrics to connect IC locking notion to intentional disruption of mixed-signal performance. We adapt and use a state-of-the-art locking technique SFLL as part of *MixLock* to enable effective trade-offs between analog and digital security, delivering a holistic protection on a given mixed-signal IC. We illustrate the application of *MixLock* on a $\Sigma\Delta$ ADC. We show that *MixLock* thwarts all known attacks in the digital domain while delivering perfect analog security levels. This is achieved without degrading the mixed-signal performance and at very reasonable area and power overheads.

ACKNOWLEDGMENTS

This work has been carried out in the framework of the ANR STEALTH project with N° ANR-17-CE24-0022-01. It is partially funded by the ANR TOLTECA project with N° ANR-16-CE04-0013-01. J. Leonhard has a fellowship from the doctoral school EDITE de Paris.

REFERENCES

[1] U. Guin et al., "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.
[2] S. Bhunia et al., "Hardware Trojan attacks: Threat Analysis and Countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.
[3] K. Tiri and I. Verbauwhede, "A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs," in *Design, Automation & Test in Europe*, 2005, pp. 58–63.
[4] A. Barenghi et al., "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.
[5] R. Torrance and D. James, "The State-of-the-Art in Semiconductor Reverse Engineering," in *IEEE/ACM Design Automation Conference*, 2011, pp. 333–338.
[6] I. Polian, "Security Aspects of Analog and Mixed-Signal Circuits," in *IEEE International Mixed-Signal Testing Workshop*, 2016.
[7] A. Antonopoulos et al., "Trusted analog/mixed- signal/RF ICs: A survey and a perspective," *IEEE Design & Test*, vol. 34, no. 6, pp. 63–76, 2017.
[8] M. M. Alam et al., "Challenges and Opportunities in Analog and Mixed Signal (AMS) Integrated Circuit (IC) Security," *Journal of Hardware and Systems Security*, vol. 2, no. 1, pp. 15–32, 2018.
[9] M. Yasin et al., "Provably-secure logic locking: From theory to practice," in *ACM/SIGSAC Conference on Computer & Communications Security*, 2017, pp. 1601–1618.
[10] D. H. K. Hoe et al., "Towards secure analog designs: A secure sense amplifier using memristors," in *IEEE Computer Society Annual Symposium on VLSI*, 2014.
[11] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *IEEE Latin American Test Symposium*, 2017.
[12] J. Wang et al., "Thwarting analog IC piracy via combinational locking," in *IEEE International Test Conference*, 2017.
[13] Maxim Integrated, "MAX36051: DeepCover Security Manager with 128 Bytes of Nonimprinting Memory," .
[14] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *USENIX Security Symposium*, 2007.
[15] C. Herder et al., "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.
[16] X. Zhang and M. Tehranipoor, "Design of on-chip lightweight sensors for effective detection of recycled ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1016–1029, 2014.
[17] J.A. Roy et al., "Ending Piracy of Integrated Circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.
[18] J. Rajendran et al., "Fault Analysis-Based Logic Encryption," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 410–424, 2015.
[19] M. Yasin et al., "On Improving the Security of Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1411–1424, 2016.
[20] P. Subramanyan et al., "Evaluating the Security of Logic Encryption Algorithms," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2015, pp. 137–143.
[21] M. Yasin et al., "SARLock: SAT Attack Resistant Logic Locking," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2016, pp. 236–241.
[22] Y. Xie and A. Srivastava, "Mitigating SAT Attack on Logic Locking," in *International Conference on Cryptographic Hardware and Embedded Systems*, 2016, pp. 127–146.
[23] M. Yasin et al., "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, 2017.
[24] M. Yasin et al., "SARLock: SAT attack resistant logic locking," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2016.
[25] K. Shamsi et al., "AppSAT: Approximately Deobfuscating Integrated Circuits," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2017, pp. 95–100.
[26] Y. Shen and H. Zhou, "Double dip: Re-evaluating security of logic encryption algorithms," Cryptology ePrint Archive, Report 2017/290, 2017.
[27] D. Haghighitalab et al., "A 2.4 GHz ISM-band highly digitized receiver based on a variable gain LNA and a subsampled Sigma-Delta ADC," *Analog Integrated Circuits and Signal Processing*, vol. 95, no. 2, pp. 259–270, 2018.