Securing Programmable Analog ICs Against Piracy

Mohamed Elshamy, Alhassan Sayed, Marie-Minerve Louërat, Amine Rhouni, Hassan Aboushady, Haralampos-G. Stratigopoulos

Securing Programmable Analog ICs Against Piracy

Mohamed Elshamy*, Alhassan Sayed*†, Marie-Minerve Louërat*, Amine Rhouni*,
Hassan Aboushady*, Haralampos-G. Stratigopoulos*
*Sorbonne Université, CNRS, LIP6, Paris, France
†Minia University, Minia, Egypt

***Abstract*—In this paper, we demonstrate a security approach for the class of highly-programmable analog Integrated Circuits (ICs) that can be used as a countermeasure for unauthorized chip use and piracy. The approach relies on functionality locking, i.e. a lock mechanism is introduced into the design such that unless the correct key is provided the functionality breaks. We show that for highly-programmable analog ICs the programmable fabric can naturally be used as the lock mechanism. We demonstrate the approach on a multi-standard RF receiver with configuration settings of 64-bit words.**

I. Introduction

Hardware security and trust is a topic that has attracted a lot of interest in recent years. There are various threats, including IC/IP piracy, hardware Trojans, side-channel attacks, and fault injection attacks, which can have serious implications on the technological value chain (e.g. CAD tool providers, IC/IP providers, original equipment manufacturers, and users), on governments, and on the society as a whole [1].

While hardware security and trust aspects have been extensively studied for digital circuits, the space of vulnerabilities and solutions for analog circuits is largely unexplored and little understood as of today [2], [3].

In this paper, we address the problem of analog IC/IP piracy, which includes reverse engineering and counterfeiting. Reverse engineering refers to the derivation of IC/IP proprietary information, i.e. architecture, netlist, layout, etc. It aims at reducing the attacker's technological disadvantage against the "author" of the IC/IP, gathering necessary information for producing a similar or identical IC/IP, e.g. a counterfeit, or locating the root-of-trust part of the IC/IP to steal secret information, such as cipher keys. Nowadays, there exist equipment and software tools to successfully reverse-engineer any unprotected IC/IP [4]. Counterfeiting includes cloning, recycling, overproducing, and remarking [5]. A cloned counterfeit is an IC/IP that is illegally cloned and sold as original. Cloning can be performed by an untrusted foundry or an adversary via reverse-engineering. A recycled counterfeit is a used and possibly aged IC that is illegally resold as new. Overproduced ICs are ICs that are produced by an untrusted foundry beyond the number agreed in the contract with the IC design house and are illegitimately sold in after market. Remarked ICs are failing ICs that are remarked by an untrusted test facility as passing ICs and are sold with false and forged documentation.

In particular, in this paper, we propose a security approach based on locking for the class of highly-programmable analog ICs. The approach can be used not only for prevention of unauthorized chip use, but, in addition, it can offer a strong countermeasure against reverse engineering and counterfeiting.

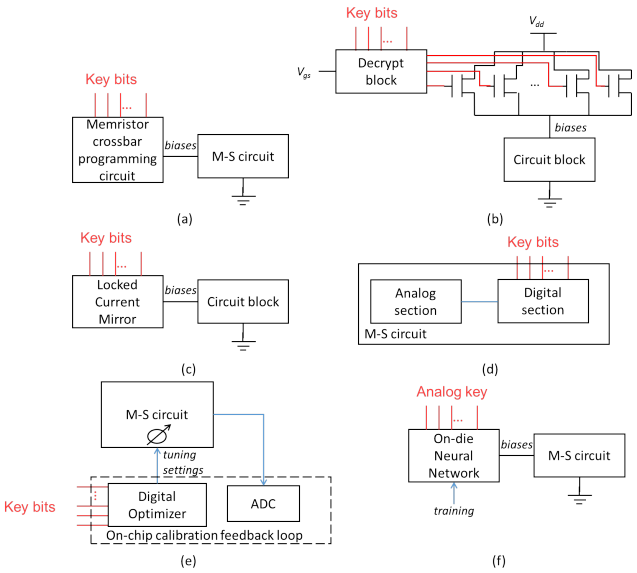

Fig. 1. Locking techniques for analog ICs: (a) locking biases based on memristor crossbars [6]; (b) obfuscating biasing transistors [7]; (c) locking of current mirrors [8]; (d) locking mixed-signal circuits via logic locking of their digital section [9]; (e) logic locking of the digital optimizer in the calibration feedback loop [10]; (f) locking through neural network-based biasing [11].

We argue that the tuning knobs within the programmable analog IC can naturally serve as a locking mechanism. Specifically, the programming bits controlling the tuning knobs serve as key-bits and each configuration setting, e.g. programming bits that configure the IC in a specific operation mode demanded by the application, is treated as a secret key. Naturally, when invalid programming bits are provided the functionality of the circuit breaks, that is, its functionality is "locked". We discuss the practical implementation of this security approach, its benefits compared to existing locking techniques, and its resilience against foreseen attacks. We demonstrate it on a highly-programmable, multi-standard RF receiver with configuration settings of 64-bit words.

The rest of the paper is structured as follows. In Section II, we discuss previous work on analog circuit locking. In Section III, we provide a general overview of programmability embedded into analog ICs. In Section IV, we present the proposed technique for securing programmable analog ICs. In Section V, we present our case study. In Section VI, we demonstrate the efficiency of locking and the achieved security level. Section VII concludes the paper.

II. Previous Work on Analog Circuit Locking

Techniques for locking of analog circuits are proposed in [6]–[11] and are illustrated in Fig. 1. In [6], a locking mechanism based on an architecture that comprises memristor

crossbars is used to lock the body biasing of the transistor input pair in a sense amplifier. In [7], it is proposed to replace transistors within the biasing circuit with parallel-connected transistors whose gates are controlled by key-bits. The key-bits set an aggregate width equal to the width of the original transistor. In [8], it is shown how to redesign the current mirrors providing the biasing so as to insert key-bits. In [9], it is proposed to lock a mixed-signal circuit via logic locking of its digital section. This technique was demonstrated in an audio application in [12] allowing to listen to the effect of locking on the audio quality. In [10], it is proposed to lock via logic locking the digital optimizer into the calibration feedback loop, such that the wrong tuning settings are generated unless the valid key is applied. In [11], it is proposed to add on-chip a neural network that is trained to map the secret analog key, which is in the form of analog DC voltages presented as inputs to the neural network, to the correct biases.

The locking approaches in [6]–[8], [11] act on the biasing of the circuit. This makes them vulnerable to removal attacks since the attacker does not have to recover the key; it suffices to recover the biases, which are typically small in number, and thereafter replace the locked blocks with "fresh ones" that provide the correct biases.

In contrast to biases that are fixed for every fabricated chip, tuning knobs are set per fabricated chip to compensate for process variations, thus to break the technique in [10] the attacker will need to actually search for the secret key. The same holds for the technique in [9] which does not lock the biasing or tuning knobs, but locks directly the functionality. Still, [9], [10] are vulnerable to removal attacks as the attacker may replace the locked digital optimizer in [10] or the locked digital section in [9] with "fresh" unlocked designs, although arguably this is a much more difficult task than redesigning the biasing circuitry.

III. Programmable Analog ICs

Analog circuits are often made programmable (or configurable) with the aim to: (a) Compensate for process variations so as to increase yield; (b) Compensate for inherent non-idealities so as to achieve the desired performance trade-off; (c) Configure the circuit into different operation modes demanded by the application; (d) Adapt the performances to changes in the environment.

The calibration mechanism that enables programmability (or configuration) consists of tuning knobs judiciously inserted into the design and a calibration algorithm that is driven by performance indicators and returns the programming bits (or configuration setting). Often the same calibration mechanism is used to achieve simultaneously multiple of the above objectives. Typically process variations are taken into consideration during calibration, thus the configuration settings end up being unique for each chip.

The calibration algorithm can either run on-chip in hardware pointing to autonomous self-calibration or can run off-chip in software during the post-manufacturing testing phase, relying on external Automated Test Equipment (ATE).

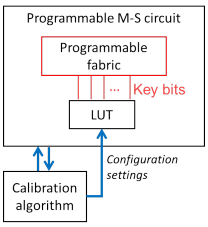

Fig. 2. Locking via the programming fabric.

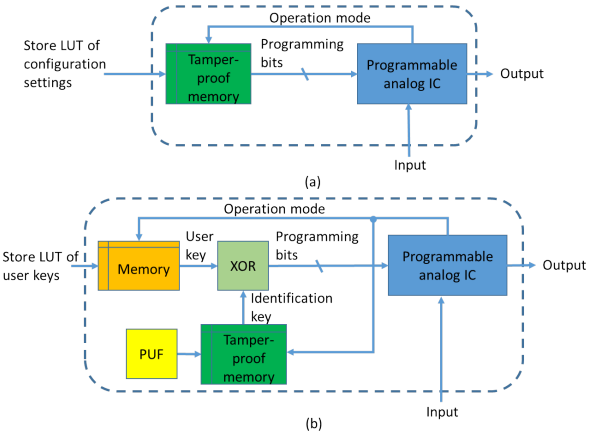

Fig. 3. Key management schemes.

For objectives (c)-(d), operation modes and adaptation levels are pre-specified based on the anticipated range of applications and environmental conditions, resulting in multiple pre-specified configuration settings that are pre-loaded into an on-chip look-up table (LUT). Instead, for objectives (a)-(b), the calibration algorithm returns a single best possible configuration setting and the LUT has essentially one line.

The above discussion remains quite general; in fact, the calibration mechanism varies from one circuit class to another, and programmability may vary from a few bits for calibrating single blocks [13] to tens of bits for calibrating complete systems [14].

IV. Securing Highly-Programmable Analog ICs

A. Locking via the programmability fabric

In this work, we argue that for highly-programmable circuits it is not required to insert additional circuitry on-chip in order to introduce key-bits. Instead, we can take advantage of the embedded programmable fabric so as to naturally perform the locking operation, as shown in Fig. 2. Specifically, the configuration settings can be treated as secret keys or, equivalently, the programming bits can be treated as secret key-bits. Similarly, the calibration algorithm that produces the configurations settings is kept secret and is not shared with untrusted parties. Using invalid programming bits will result in complete loss of functionality or in significant performance degradation, that is, one or more performances will lie far outside their allowable specification range.

Exploiting the embedded programmability for performing the locking operation presents significant advantages. Unlike known approaches for locking analog ICs which all modify to some degree the design [6]–[11], with the proposed approach the design is left completely intact. Therefore, there is no

need for redesign, no extra design iterations, and, most importantly, no performance degradation. In addition, the proposed approach does not increase the power or area of the analog IC itself. The power and area overheads are only due to the key management scheme, which can be shared for enabling security for all other blocks on the same die, i.e. in the context of a System-on-Chip (SoC).

The possible secret key management schemes are the same ones used by logic locking techniques for digital ICs [15]. One option is to store the LUT with the configuration settings into a tamper-proof memory, as shown in Fig. 3(a). A second option, illustrated in Fig. 3(b), makes use of a Physical Unclonable Function (PUF) [16]. The PUF needs to take at least as many challenges as the total number of configuration settings. The PUF generates a number of secret identification keys that equals the number of configuration settings. The user is given a number of keys that also equals the number of configuration settings, such that when these keys are XORed with the identification keys the correct configuration settings are produced. In both schemes, in normal operation mode the circuit commands dynamically the memories to load the corresponding programming bits.

B. Attack scenarios and resilience analysis

We assume that an attacker has full capabilities, i.e., has the netlist and access to working oracle chips.

1) Attacks in digital domain: Known attacks in digital domain, such as the lethal SAT attack [17], are not applicable.

2) Removal attacks: Removal attacks, which is the main limitation in [6]–[11] as described in Section II, are not applicable as there is no added circuitry on-chip to facilitate the key insertion; the key directly applies to existing tuning knobs into the design.

3) Brute-force and multi-objective optimization attacks: The most trivial attack is the brute-force attack which consists in applying random combinations of programming bits until the one that unlocks the circuit is found.

The multi-objective optimization attack consists in applying an iterative algorithm that searches for a configuration setting that simultaneously optimizes the performances such that they all satisfy their specifications. This attack is difficult to put in place since typically only a small subset of programming bits shows a smooth monotonic relationship with a given performance, and this requires that the rest of the programming bits are already correctly set.

Performing the brute-force and multi-objective optimization attacks by simulation is impractical due to very long analog simulation times. To perform these attacks much faster in hardware, the attacker needs first to re-fab the circuit so as to gain direct access to programming bits. In this case, if the programming bits are unique for each chip, then these attacks become meaningful only if the resultant key-bit combination can be used to set a good starting point for launching a gradient search for quickly calibrating any chip.

A question rises whether the design can be divided in sub-blocks, tracing key bits to sub-blocks, and enabling smaller brute-force and multi-objective optimization attacks at sub-block level. This is typically not possible due to the internal feedback loops that involve multiple sub-blocks each. Moreover, this would require re-fabbing so as to provide intermediate taps to access directly sub-blocks for measurement, which is only possible for low-frequency sub-blocks.

It should be noted that unlike logic locking of digital circuits where there is a single valid key, for analog circuits it is likely that a number of key-bit combinations result in a satisfactory performance trade-off, although this number is typically a very small fraction of all key-bit combinations.

Clearly, resilience against these attacks increases with the number of programming bits and with simulation or measurement time per trial.

4) Revealing the calibration algorithm: In the case of an on-chip calibration, an attacker that extracts the netlist will also have at hand the hardware that implements the calibration feedback loop and, thereby, it may be fairly easy to extract the calibration algorithm. In this scenario, we can envision logic locking of the digital section of the calibration feedback loop [9], [10].

In the case of an off-chip calibration, the attacker may target speculating the calibration algorithm by studying the circuit architecture. However, very often the calibration algorithm is very specific and esoteric to the design, that is, intended for or understood only by the designer. Thus, the attacker must have a very high and specialized expertise and a thorough understanding of the design so as to be able to conceive the underlying calibration algorithm. This is in contrast to the security assumption, in the sense that if the attacker is so knowledgeable, then the attacker may as well design the circuit from scratch instead of pirating an existing one, given also that piracy by an attacker other than the foundry requires a sophisticated and expensive reverse-engineering infrastructure. This is a new type of attack specific to the countermeasure that is proposed. In general, it opens a discussion of securing and obfuscating calibration algorithms when they are considered to be a valuable intellectual property of the design. A metric to quantify the difficulty for reverse-engineering a calibration algorithm will need to devised also.

A problem that needs to be addressed is how to protect the calibration algorithm against an untrusted test facility. For programmable analog circuits, multiple test/calibration iterations are carried out during testing for searching in the space of configuration settings. The search is not random, but is driven by the calibration algorithm. In each step of the algorithm, the next configuration setting is dictated by tests done using the current configuration setting. If the calibration runs entirely on-chip, then it is transparent to the test facility. If the calibration runs off-chip, then the obvious workaround is that the foundry returns the manufactured chips to the design house, where the test/calibration steps are performed in a secured environment. However, this option is practical only in the case of low-volume products. For high-volume products, it is straightforward to adapt the concept of remotely activating the chips using asymmetric cryptography [15].

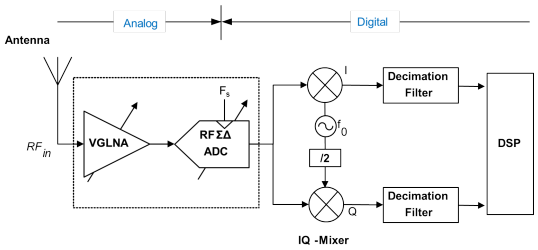

Fig. 4. Architecture of programmable multi-standard RF receiver.

C. Threats being addressed

Assuming resilience against the aforementioned attacks, the proposed approach can be used as countermeasure for reverse-engineering, cloning, overproducing, and remarking. It can offer resilience against recycling only if the key management scheme in Fig. 3(b) is used and the unique user keys are loaded every time at power-on. More specifically, a cloned counterfeit that can be produced by reverse-engineering a chip or by the untrusted foundry is good-for-nothing if the adversary does not know how the design can be programmed. Resilience against overproducing can be achieved since the design house can have control over the number of correctly programmed fabricated chips. Remarking can be achieved since the design house can load after unsuccessful calibration wrong configuration settings to render the chip totally malfunctional.

V. Case Study: Programmable Multi-Standard RF Receiver

Our case study is a programmable highly-digitized multi-standard 65 nm CMOS RF receiver. It is made re-configurable such that it can serve for establishing communication using several standards within the frequency range from 1.5 GHz to 3.0 GHz, including Bluetooth, ZigBee, WiFi 802.11b, etc.

A. Architecture and Programmability

Fig. 4 shows the block-level schematic of the RF receiver. It is composed of a Variable Gain Low Noise Amplifier (VGLNA), a band-pass (BP) RF $\Sigma\Delta$ modulator, a digital down-conversion mixer, and a digital decimation filter. In total, there are 64 programming bits embedded into the analog section and 3 programming bits embedded into the digital section. The calibration is run off-chip and returns the programming bits per standard in the presence of process variations, thus resulting in unique configuration settings per standard and per chip. For the purpose of locking, we consider only the programming bits of the analog section since the calibration of the digital section for a given standard is straightforward.

The block-level schematic of the VGLNA is shown in Fig. 5. It is composed of five gain stages with a resistive feedback. It features a 4-bit configuration word with which the VGLNA can attain 16 different gain levels so as to adapt the sensitivity and dynamic range of the RF receiver to the specifications imposed by the target standard.

The block-level schematic of the BP RF $\Sigma\Delta$ modulator is illustrated in Fig. 6 [18]. It is composed of an input transconductance, Gm_{in}, an LC bandpass loop filter with two capacitor arrays C_c and C_f for coarse- and fine-tuning,

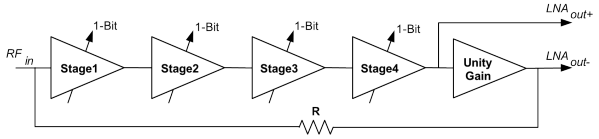

Fig. 5. Architecture of tunable variable gain LNA.

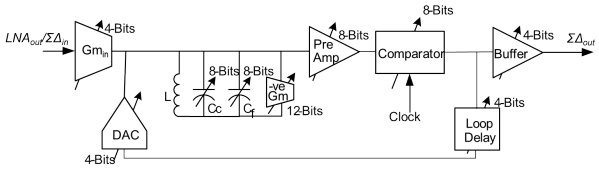

Fig. 6. Architecture of tunable BP RF $\Sigma\Delta$ modulator.

respectively, a pre-amplifier, a comparator, a loop delay, a feedback Digital-to-Analog Converter (DAC), and an output buffer. For a target standard, the modulator uses a 60-bit configuration word to tune the center frequency and quality factor of the LC bandpass loop filter in the presence of process variations, as well as to trim the biasing current of the other blocks in order to compensate for process variations and improve the performance trade-off.

B. Calibration Algorithm

The calibration procedure for a target standard is as follows:

1) The comparator is configured as a buffer by deactivating its driving clock.
2) The output buffer is configured to adapt the output of the BP RF $\Sigma\Delta$ modulator to its off-chip load during calibration. This output buffer is removed from the signal path in normal operation mode.
3) The RF input signal is disabled by turning off the input transconductor Gm_{in}.
4) The feedback loop with the DAC and loop delay is turned off.
5) Having deactivated the feedback loop, the LC loop filter is put in oscillation mode by setting its Q-enhancement transconductor, -ve Gm, to its maximum.
6) The capacitor arrays C_c and C_f of the LC tank are tuned until the output frequency is equal to the desired center frequency.
7) The Q-enhancement transconductor, -ve Gm, is reduced gradually until oscillation vanishes.
8) The feedback loop is restored.
9) The BP RF $\Sigma\Delta$ modulator is put in the operating mode by applying an RF input signal with frequency F_0.
10) The sampling frequency is set to $F_s = 4 \cdot F_0$.
11) The loop delay is set according to F_s.
12) The VGLNA is tuned to set the appropriate sensitivity and dynamic range.
13) The input transconductance Gm_{in}, the feedback DAC, the pre-amplifier and the comparator are initialized to their nominal values determined by simulation.
14) An iterative procedure is used to determine the configuration words of these blocks through the improvement of the measured Signal-to-Noise Ratio (SNR) and Spurious Free Dynamic Range (SFDR) of the BP RF $\Sigma\Delta$ modulator.

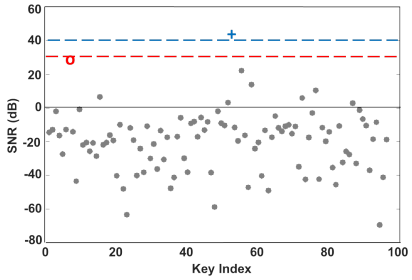

Fig. 7. SNR for correct key (blue cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the BP RF ΣΔ modulator.

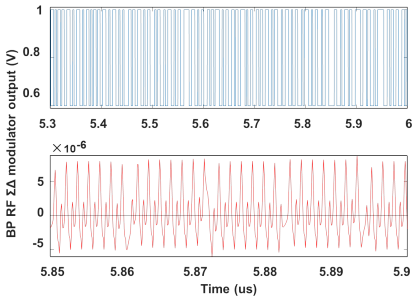

Fig. 8. Transient output of BP RF ΣΔ modulator for the correct key (top) and the invalid key with index 7 in Fig. 7 (bottom).

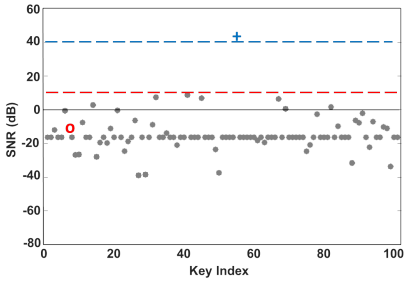

Fig. 9. SNR for correct key (blue cross) and invalid keys (gray dots and red dot with index 7) computed at the output of the RF receiver.

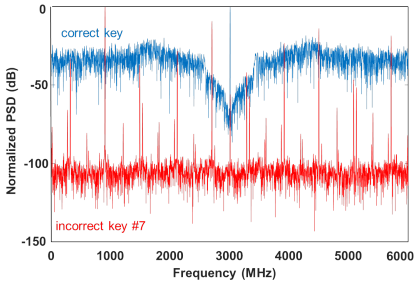

Fig. 10. PSD at the output of the BP RF ΣΔ modulator for the correct key (blue) and the invalid key with index 7 in Figs. 7 and 9 (red).

VI. Results

A. Locking efficiency

For a given center frequency or standard, there is an optimal combination of the 64 programming bits composing a secret key that unlocks functionality. We will consider the maximum center frequency, e.g. 3 GHz, and we will demonstrate the locking efficiency when applying invalid keys. The circuit has several performances, including SNR, dynamic range, SFDR, etc., and locking succeeds when at least one performance violates its specification.

We assume that the attacker has extracted the netlist of the circuit and can simulate it at transistor-level with the ability to monitor internal nodes that shed more light into the operation. We consider first the SNR observed at the output of the BP RF ΣΔ modulator for an input sinusoidal signal with frequency 3 GHz and power -25 dBm. The SNR is computed for an Oversampling Ratio (OSR) of 64 and based on a 8192 point FFT. Fig. 7 shows the SNR across 100 randomly generated keys and the correct key. As it can be seen, the correct key stands out resulting in an SNR of over 40 dB, while for invalid keys the SNR is less than 30 dB. In fact, for most invalid keys the SNR is below 0 dB, which means that the input signal gets buried under the noise level or there are harmonics within the band-of-interest. However, there are 4 invalid keys that have an SNR higher than 10 dB, and among them one has an SNR of about 30 dB. This key with index 7 in Fig. 7 naturally would attract the attention of the attacker. But, in fact, it turns out to result in a "deceptive" SNR. Looking into the programming bits corresponding to this key, the feedback loop of the BP RF ΣΔ modulator is open and, in addition, the comparator operates as a buffer. In this way, the analog signal passes without being digitized, thus there is no quantization noise being added. Fig. 8 shows the transient output of the BP RF ΣΔ modulator for the correct key and the invalid key with index 7. The correct output is an oversampled bitstream, while the output for the invalid key is an analog waveform showing no analog-to-digital conversion. This analog waveform when it passes directly through the digital section of the RF receiver will show a reduced SNR. This is shown in Fig. 9, where now the SNR measured at the output of the RF receiver is plotted. The SNR for the correct key does not change as expected and for some invalid keys the SNR is further reduced. In short, all invalid keys show an SNR of less than 10 dB, that is, the functionality is significantly corrupted.

Fig. 10 shows the Power Spectral Density (PSD) at the output of the BP RF ΣΔ modulator for the correct key and the invalid key with index 7. As it can be seen, for the invalid key there is no noise shaping, which is the main characteristic of the BP RF ΣΔ modulator.

Fig. 11 shows the dynamic range of the RF receiver for the correct key and the invalid key with index 7. The input range is divided into three segments, e.g. [-85:-45], [-60:-20], and [-40:0], and for each segment the VGLNA is tuned to have the appropriate gain level and sensitivity. While Figs. 7 and 9 show the SNR just for an input power of -25 dBm, Fig. 11 plots the SNR for different input power values with a step of 5 dBm. As it can be seen, the behavior of the locked circuit across the input range is very different as compared to the unlocked circuit.

Fig. 12 shows the SFDR for the correct key and the invalid key with index 7. SFDR is measured by applying a two-tone input, where the two tones have the same power and a frequency difference of 10 MHz. SFDR is the difference between the power of the fundamental and the third harmonic. As it can be seen, the locked circuit has a much lower SFDR.

Finally, the same experiment was repeated for other center

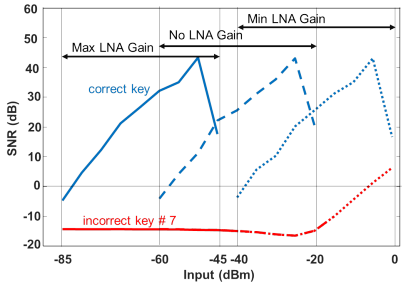

Fig. 11. SNR versus input power with different LNA gain settings for the correct key (blue) and the invalid key with index 7 in figures 7 and 9 (red).

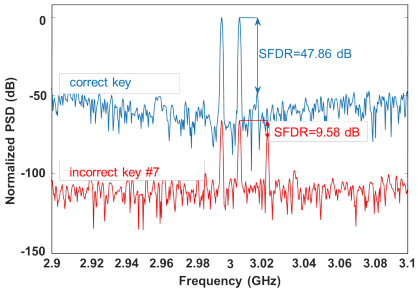

Fig. 12. SFDR for the correct key (blue) and the invalid key with index 7 in Figs. 7 and 9 (red).

frequencies and qualitatively the results were identical.

B. Security Analysis

1) Brute-force and multi-objective optimization attacks: Simulation of this device is too time-consuming, thus the attacker will need to re-fab the chip so as to perform the analysis in hardware. For example, for a single key and a 8192 point FFT, it takes about 20 minutes to simulate the SNR at the output of the RF receiver for a given input, 3 hours to simulate the SNR across the input range, and 30 minutes to simulate the SFDR.

Most sub-blocks are included in a feedback loop which makes it impossible to calibrate individual sub-blocks. Also, to calibrate a sub-block, the rest of the sub-blocks need to be conditioned appropriately. Besides, the circuit carries high-frequency signals and re-fabbing a chip with intermediate taps for observing internal signals would result in performance loss.

Resilience against these attacks is also naturally achieved thanks to the large key-width of 64 bits which explodes the search space. It is very unlikely that many key-bit combinations could result in satisfactory performance trade-off. For example, capacitor arrays are binary-weighted, thus for a desired capacitor value there is a unique sub-key.

2) Speculating the calibration algorithm: The calibration procedure is arguably very complex and its steps cannot be easily retraced by conjecture even under the assumption that the attacker has strong analog design expertise. In particular: (a) The circuit needs to be reconfigured appropriately multiple times during calibration; (b) The calibration of many sub-blocks requires initial programming bits that are dictated by design simulation and are unknown to the attacker; (c) The order with which the different blocks should be calibrated is very specific; (d) The feedback loop prohibits calibration of individual sub-blocks.

VII. Conclusion

We propose a locking methodology for highly-programmable analog ICs that relies on securing the configuration settings as well as the underlying calibration algorithm. The methodology can be used as a countermeasure for unauthorized chip use and piracy. It is demonstrated on a multi-standard RF receiver with 64 programming bits. The proposed methodology has the advantage that it is strictly non-intrusive to the design and has zero power and area overheads, except for the overheads due to the key management scheme.

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.

References

[1] M. Rostami et al., "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.

[2] 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.

[3] 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.

[4] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *IEEE/ACM Design Automation Conference*, 2011, pp. 333–338.

[5] 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.

[6] 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.

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

[8] J. Wang et al., "Thwarting analog IC piracy via combinational locking," in *IEEE International Test Conference*, 2017.

[9] J. Leonhard et al., "Mixlock: Securing mixed-signal circuits via logic locking," in *Design, Automation & Test in Europe Conference*, 2019.

[10] N. G. Jayasankaran et al., "Towards provably-secure analog and mixed-signal locking against overproduction," in *IEEE/ACM International Conference on Computer-Aided Design*, 2018.

[11] G. Volanis et al., "Analog performance locking through neural network-based biasing," in *IEEE VLSI Test Symposium*, 2019.

[12] J. Leonhard et al., "Mixed-signal hardwre security using mixlock: Demonstration in an audio application," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019.

[13] M. Andraud et al., "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016.

[14] S. Li et al., "Reconfigurable All-Band RF CMOS Transceiver for GPS/GLONASS/Galileo/Beidou With Digitally Assisted Calibration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1814–1827, 2015.

[15] J. A. Roy et al., "Ending piracy of integrated circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.

[16] C. Herder et al., "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[17] P. Subramanyan et al., "Evaluating the security of logic encryption algorithms," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2015.

[18] A. Ashry and H. Aboushady, "A 4th order 3.6GS/s RF Sigma-Delta ADC with a FoM of 1pJ/bit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2606–2617, 2013.