The Key is Left under the Mat: On the Inappropriate Security Assumption of Logic Locking Schemes

M Tanjidur Rahman, Shahin Tajik, M Sazadur Rahman, Mark Tehranipoor, and Navid Asadizanjani
Florida Institute for Cybersecurity (FICS) Research
ECE Department, University of Florida
{mir.rahman, stajik, mohammad.rahman}@ufl.edu, {tehranipoor, nasadi}@ece.ufl.edu

***Abstract*—Logic locking has been proposed as an obfuscation technique to protect outsourced IC designs from IP piracy by untrusted entities in the design and fabrication process. In this case, the netlist is locked by adding extra key-gates, and will be unlocked only if a correct key is applied to the key-gates. The key is assumed to be written into a non-volatile memory after the fabrication by the IP owner. In the past several years, the focus of the research community has been mostly on Oracle-guided attacks, such as SAT attacks, on logic locking and proposing proper countermeasures against such attacks. However, none of the reported research in the literature has ever challenged a more fundamental assumption of logic locking, which is the security of the key itself. In other words, if an adversary can read out the correct key after insertion, the security of the entire scheme is broken. In this work, we first review possible adversaries for the locked circuits and their capabilities. Afterward, we demonstrate that even with the assumption of having a *tamper-* and *read-proof* memory for the key storage, which is not vulnerable to any physical attacks, the key transfer between the memory and the key-gates through registers and buffers make the key extraction by an adversary possible. To support our claim, we implemented a proof-of-concept locked circuit as well as one of the standard logic locking benchmarks on an FPGA manufactured with a 28 nm technology and extract obfuscation keys using optical probing. Finally, we discuss the feasibility of the proposed attack in different scenarios and propose potential countermeasures.**

***Index Terms*—Logic Locking, Optical Probing, Tamper-proof Memory**

I. Introduction

The supply chain of integrated circuits (ICs) has changed significantly over the past two decades. The globalization of semiconductor manufacturing has been on the rise due to the high demand for smaller technologies, reduction in manufacturing cost, and shortened time-to-market. Hence, the business model for the semiconductor industry has shifted from the vertical model towards the horizontal model, see Fig. 1. In the horizontal model, different steps of chip manufacturing, such as design, integration, fabrication, and packaging may no longer be completed under the same roof. In this case, with many entities involved in the supply chain that are located across the globe, original IP owners no longer have control over the entire supply chain [1], [2]. Hence, Intellectual Property (IP) vendors and design houses are facing the threat of IP theft/piracy, tampering, overproduction, and counterfeiting. In the past years, IP protection was entirely dependent on passive protection schemes like patents, copyrights, and watermarks. Due to the failure of the protections mentioned above schemes, researchers have focused on developing active approaches like IC metering [3], IP encryption [4], logic locking/ obfuscation [5], [6], state space obfuscation [7], secure split manufacturing [8], and IC camouflaging [9].

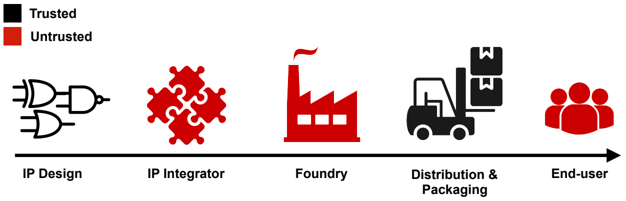


Fig. 1: Horizontal model of supply chain for the semiconductor industry.

Among the aforementioned solutions, logic locking is emerging as a possible solution for establishing trust in the hardware design. Logic locking is a method of protecting the confidentiality of IP by locking the original circuitry using additional logic elements like XOR gates or multiplexers into the IP. The locking logic elements are generally termed as key-gates. The circuit is unlocked if the IP receives a correct key configured by the IP owner through a non-volatile memory (NVM) after the chip is fabricated. Although logic obfuscation appeared as a promising protection mechanism against IP piracy, the literature shows that it is vulnerable to Boolean satisfiability (SAT) attacks [10], [11], signal probability skew attacks [12], bypass attacks [13], and key sensitization attacks [14]. These attacks are mostly dependent on the analysis of input/output patterns received from an unlocked chip, and hence, they are referred to as Oracle-guided attacks. While protection against the above-mentioned attacks received so much attention [15]–[17], unfortunately, no attention has been given to the security of the key itself. The reason behind overlooking the security of the key is lying under two common assumptions made by all those attacks. First, a potential adversary is an untrusted foundry, which does not have access to the unlocking key during fabrication. Second, it is assumed that the secret key is written into a *tamper-* and *read-proof* memory, and therefore, it is protected against reverse engineering in the field. However, no prior work has evaluated the validity of these assumptions.

Adversaries, such as untrusted foundries or reverse-engineering entities like Techinsights [18], are equipped with the most advanced failure analysis (FA) equipment, e.g., scanning electron microscope (SEM) or laser scanning microscopes (LSM). Hence, it is conceivable that an untrusted foundry gets access to the shipped product in the market and use their FA capabilities to extract the unlocking key from the obfuscated chips. Techniques, such as optical probing [19], [20] or microprobing [21], [22], can be employed to localize points of interests (PoI) and probe the key movement between the memory and the locked logic. Besides, the assumptions mentioned above do not consider the threat imposed by an end-user through full-blown or partial reverse engineering. If an entity is able to reverse-engineer a chip (fully or partially), that is the indicative of the fact that the entity has excellent FA capabilities. As a result it is only logical to assume that such capability could be used to attack the on-chip key. Hence, an in-depth analysis of attack models for logic locking algorithms is required to fill the voids in countermeasures proposed to secure the obfuscated IP.

Our Contribution. The primary contributions of this work are summarized as follows:

- We present the attack models for the complete life cycle of a modern chip. For this purpose, we have analyzed the information available to different adversaries, such as System on Chip (SoC) integrators, untrusted foundries, or end-users. Afterward, we evaluate the practicability of extracting key values for unlocking a locked circuitry by these adversaries, with or without having access to the circuit layout.
- We show that the assumption of having a tamper- and read-proof memory for the key storage is not sufficient for logic locking schemes. Such a memory may provide security for the chip at a power-off state, but a fully functional chip can expose the key signal on a bus or register for probing.
- To validate our claims, we conducted experiments on logic locking implementations on a Flash-based FPGA fabricated in 28 nm technology node. We demonstrate that an attacker can extract the entire key by localizing and probing the key-gates/registers using optical probing from the IC backside. Our results show that logic locking can even be vulnerable to physical attacks mounted by an end-user with no access to the circuit layout.

It should be noted that optical probing is only one of several available FA techniques, and therefore, other methods can also be used to extract the key. However, the advantages of optical probing in comparison to other techniques are two-folds, namely (i) it can localize and probe simultaneously and (ii) it can be non-invasive. Finally, we propose possible countermeasures to mitigate the shortcomings of the current logic locking schemes.

II. Background

A. Logic Locking

Logic locking has been developed as an obfuscation technique to conceal the functionality, and design of IP cores to provide protection against malevolent reverse engineering and reusing attempts. Such protection is provided through embedding additional key-controlled logic gates, known as *key-gates*, in the netlist of the IP. If the key value of the key-gates is fed through a set of registers, we call them *key-registers* throughout the paper. The correct functionality of the IP is only achieved if the correct key is fed to key-gates. The key is not available during the fabrication process and is inserted into an NVM (e.g., Flash, EEPROM, or e-fuse) inside the chip before releasing the chip into the market or deploying it into the field. Consequently, the correct functionality is hidden from an untrusted foundry during manufacturing. Since the random insertion of key-gates, without considering circuit structure, does not necessarily raise the security level of the design, several key-gate insertion algorithms have been proposed in the literature [5], [23], [24]. Besides, when obfuscating a circuit with multiple modules, each module is obfuscated with a number of key bits that may be proportional to the size of the module [25]. The locking keys are assumed stored in a secure memory. Therefore, it is considered that if the key storage is secured, the security of the IP can then be ensured by appropriate insertion of key-gates.

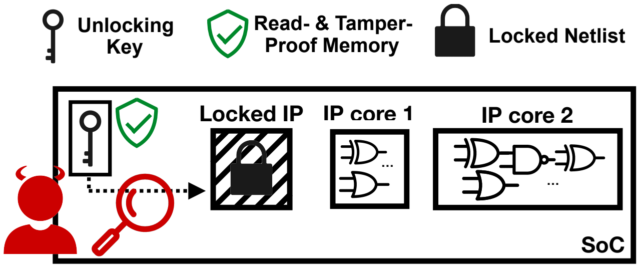


Fig. 2: Key extraction during the key transfer between the secure memory and the locked circuit.

B. Tamper- and Read-Proof Memory

The existence of a tamper- and read-proof memory is the primary assumption of the logic locking technique. It is assumed that the unlocking key can be stored after manufacturing in a secure manner that its content cannot be extracted. In fact, there are memory technologies where it is very hard to read their content, even with the most sophisticated FA tools, if no electrical interface is available to the outside world. A conventional example of such memory is the flash/EEPROM technology, where measuring the trapped charges in the floating gate of transistors is not a straightforward task [26]. In contrast to flash/EEPROM memories, other NVM technologies, e.g., eFuses, battery-backed RAMs, and ROM, are more susceptible to direct readout. For instance, the cell states of eFuses and ROMs can be observed by SEM [27], or battery-backed RAMs can be read out by optical techniques, such as thermal laser stimulation (TLS) [28]. Physically unclonable functions (PUFs) have demonstrated similar vulnerabilities to optical techniques as well [19].

However, regardless of the tamper-resiliency and security of the memory itself, the transmission of data from/to the memory still leaves the door open to an adversary to probe or tamper with the content of the memory, see Fig. 2. The movement of data through buffer and registers enables an adversary, who has access to FA tools, to localize and probe the confidential data. Naturally, established countermeasures, such as memory encryption and authentication, are also not effective in case of logic locking, since these solutions still require a secure memory to store encryption/authentication keys. Consequently, it is not sufficient to assume the existence of a secure storage. Note that the problem of secure storage is not limited to the logic locking schemes. Indeed, it is an old problem in the field of cryptographic hardware, where the secret key has to be kept confidential on the chip. The difference here, however, is that one assumes a layout/netlist is available to an adversary with significant FA capability (e.g., an untrusted foundry or a reverse-engineering entity), who should be well equipped to carry a rather straightforward non-destructive attack, such as optical probing, and easily steal the key.

C. Optical Probing

Optical FA techniques have been promoted as a solution for contactless IC debugging from the backside of a chip. Contactless interaction with the transistors requires much less effort in comparison to other debugging tools, such as Focused Ion Beam (FIB) circuit editing. Besides, the transparency of silicon to photons in near-infrared (NIR) spectrum aligned with the popularity of flip-chip packages makes the optical analysis of operating chips in a non-invasive way possible [20]. The two major optical probing techniques are electro-optical probing (EOP) and electro-optical frequency mapping (EOFM). While EOP can be used to probe electrical signals on the transistors directly EOFM can be employed to create an activity map of active circuits. In both cases, the photons pass through the silicon substrate, which leads to partial absorption and reflection of photons in the active region. In the case of EOFM, a laser scans the region of interest on the device under test (DUT) and the reflected light is fed into a spectrum analyzer acting as a narrow band frequency filter [29]. The output from spectrum analyzer is sampled for every scanned pixel and then a PC is used to assemble the sampled frequency filter values into 2D image using grayscale color representation [20]. If a node operates at the frequency of interest, it will modulate the light reflected with the same frequency. The locations of the node operating at the same frequency are identified as a bright or dark spots when the signal is fed into the spectrum analyzer

D. Reverse Engineering

Reverse Engineering has several meanings in the context of hardware security. In this work, we make a distinction between *full-blown* and *partial* reverse engineering. A complete or full-blown reverse engineering is comprised of five stages; (a) decapsulation to remove the IC package (b) delayering the bare die (c) imaging (d) annotating each element in the images and (e) extracting netlist of the chip [30], [31]. Through full-blown reverse engineering complete the extraction of layout, netlist, and functionality of IC is possible. On the other hand, obtaining information about the operation and functionality of the chip without exposing the RTL netlist are defined as partial reverse engineering. For instance, side-channel leakages, such as electromagnetic radiation, power leakage, and photon emission, expose sensitive information about chip operation and functionality.

TABLE I: A comprehensive attack model for logic locking

Attacker	**Asset Holding**	**Challenges**	**Advantages**
Untrusted Foundry or	GDSII, Unlocked chip	Localizing key-gates/register	Access to locked layout
SOC Integrator	Soft/hard IP, Unlocked chip	Localizing key-gates/registers	1. Access to locked netlist and layout 2. Knowledge of IP functionality
End User	Unlocked chip	complete/partial reverse engineering for layout and netlist extraction	1. Knowledge of I/O pin configuration 2. Access to chip datasheet

III. Potential Adversaries

In this section, we describe all possible circumstances in which a vulnerability can be exploited by a potential adversary during the complete life-cycle of the chip. Our attack method is motivated by the fact that the logic locking key is stored in an NVM (e.g., flash or eFuses). During the bootup of any chip, to avoid glitches and latency due to key reading form NVM, the key values are transferred from the memory to key-gates through registers [32]. Therefore, localizing those key carrying registers or gates can provide suitable locations for probing the data to extract the input sequence required for those key-gates.

In the semiconductor industry, the involvement of 3rd party entities in fabrication, packaging assembly and distribution process does not leave the scope for a fully trusted supply chain (Fig. 1). Therefore, while developing the attack models, our center of attention is to include all possible adversaries in the supply chain. Eventually, available resources, advantages, and challenges for those adversarial entities are analyzed. The assets, challenges, and advantages available to each adversary are summarized in Table I.

A. Untrusted Foundry

A foundry has access to state-of-art reverse engineering and failure analysis tools. Besides, it has access to physical layout and GDSII file of the design intended for fabrication. With such access to advanced tools and confidential information, an untrusted foundry becomes a potential antagonist for IP confidentiality. A malicious foundry can reverse engineer the IP core from the GDSII file and localize the key-gates and key-registers. In addition to the layout information, the attacker can also obtain activated chips, i.e., the chip which can return correct output for any input pattern. Such an IC can be obtained from the open market, a malicious insider in trusted entities in the supply chain, or from a fielded system.

B. SoC Integrator

An SoC integrator has access to the hard/soft IP core as well as knowledge about the functionality of the chip. Besides, she can get access to the unlocked chip available in the open market. The SoC integrator might have a similar motivation, like untrusted foundry, for IP piracy through unlocking the IP functionality.

C. End-User and Reverse Engineering Entity

An end-user can have permanent/temporary access to reverse engineering capabilities similar to facilities available to a foundry. In this case, she does not necessarily have access to the layout and GDSII file. However, she can gain knowledge of

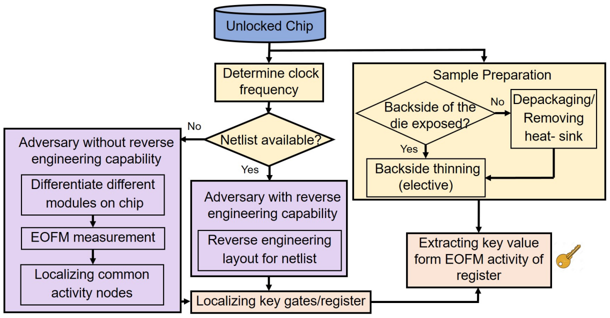

Unlocked Chip
Determine clock frequency
Sample Preparation
Backside of the die exposed?
No
Depackaging/ Removing heat- sink
Yes
Backside thinning (elective)
No
Netlist available?
Yes
Adversary without reverse engineering capability
Differentiate different modules on chip
EOFM measurement
Localizing common activity nodes
Adversary with reverse engineering capability
Reverse engineering layout for netlist
Extracting key value form EOFM activity of register
Localizing key gates/register

Fig. 3: Simplified illustration of key extraction methodology from logic obfuscated circuitry using optical contactless probing.

key-gate/register location through complete netlist extraction by reverse engineering of the chip. Moreover, from the available documentation along with access to the unlocked chip, the adversary can learn about the functionality of the chip. The motivation for the end-user is also similar to the untrusted foundry. This is possibly the most inexpensive case and the most dangerous one, where a single person can potentially be the attacker. This kind of adversary is the primary focus of this paper.

IV. Attack Approach

In this section, we present how an attacker can proceed to break into a logic locking circuitry using optical probing. For a successful attack against logic locking involving malicious foundries or SoC integrators, or end-users with reverse engineering capabilities, we assume the following information is available. First, the GDSII layout of a logic obfuscated chip or IP is available. Hence, the attacker can partially or entirely reverse engineer the chip or an IP, and thus, localize the registers or key-gates. Second, an attacker has access to an unlocked IC and has knowledge about the functionality of the chip. Third, the attacker has access to an optical probing system; such a system is available in any FA lab and can be rented for a couple of hundred dollars per hour. In addition to that, she may need standard lab equipment, which are available in the market.

Among these three assumptions, the first one is not applicable to an end-user who does not have the full-blown reverse engineering capability. To find out the key value form a target IP using optical probing, she would need to complete the three following steps: a) Preparing the sample for probing; b) Partial reverse engineering the chip to localize the key-registers; c) Extracting the key value form key-gates/registers. The steps for extracting key values are shown in Fig. 3.

A. Sample Preparation

Non-flip-chip ICs, i.e., known as wire bond chips, are required to be depackaged for analysis from the backside (Fig. 3). Besides, the chip must be operational after exposing the backside of the die for EOFM analysis. However, most modern chips are available in flip-chip packages, where the silicon substrate on the backside of the chip is usually covered with a heat-sink. In this case, the removal of the heat sink exposes the silicon on the backside of the chip, and thus, the chip is ready for optical analysis. The adversary can deploy X-ray imaging [33] for localizing the die under the heat sink, and ensuring the integrity of the die during the heat-sink/package removal. Using hotplate and lab knife the heat-sink over the chip can be removed easily. For the ICs other than flip-chip packaging, acid etching or selective mechanical polishing can be used to expose the backside of the die. Once the backside of the chip is exposed, the attacker can use further selective polishing to increase the resolution of the laser scanning image. However, this step might not be necessary since the modern optical probing system has the capability to change the depth-of-focus of the microscope depending on the thickness of silicon backside.

B. Determining Clock Frequency

The registers in an IC are connected to the clock tree for a standard chip architecture. Thus, for an end-user who does not have knowledge about the location of key-registers, the clock frequency plays a crucial role for revealing the location of sequential logic elements on the chip using EOFM. The attacker can determine the clock frequency by analyzing the documentation available for the chip to specify the frequency value for the chip. For an untrusted foundry, electromagnetic and power side-channel in frequency domain can assist to detect the exact frequency of the internal clock.

C. Reverse Engineering and Localizing Key-Gates/Register

The method of localizing key-gates/registers depends on the availability of the layout and capabilities available for the adversary. An adversary, like an untrusted foundry, an SoC integrator or a reverse engineer, can uncover the location of key-gates and key-registers by analyzing the GDSII or performing full-blown reverse engineering. On the contrary, an end-user without full-blown reverse engineering capability can focus on partial reverse engineering of the chip using publicly available documents and side-channel analysis of unlocked chip to reveal the key-register location.

The end-user without having the layout can initiate the partial reverse engineering by analyzing the image of backside of the complete die. This step is pretty straight forward if she has access to an optical probing system. She can acquire reflected light images of the complete die with a 1.3μm laser beam. Silicon is transparent to this wavelength, which can deliver images of circuit structure lying beyond the silicon. As a result, she can distinguish between different modules on the chip, such as memory blocks and logic areas. The memory and cache blocks are consisting of repetitive features which can be identified from reflected light images. Logic areas, on the other hand, are composed of different blocks for individual sub-functions and synthesized logic areas, and therefore, possess a more irregular structure. The logic area can be considered as the possible location of key-gates/registers.

To reveal the exact location of key-register without access to the layout, an end-user can focus on measuring the EOFM activity during the bootup process. During the bootup process, the ICs initiate the keys required for security modules like cryptographic cores. For modern processors, the booting keys are embedded in one-time programmable (OTP) memory or secure memory [34], [35]. These keys are used for authenticating the operating system [36], [37]. Such a bootup process is widely known as secure bootup in industry. The secure bootup process initiates the secure communication between hardware and firmware with the outside world and establishes a secure environment for the functionality of the chip. Since the logic locking keys are imperative to the functionality and security of the chip, the keys should be loaded at key-registers during the bootup process. Therefore, once the locking key is read from the memory, it is fed to the registers or flip-flops connected to the key-gates distributed all over the chip [38]. These registers should be privileged registers to prevent any inadvertent manipulation of key values and should maintain the stored data throughout the operational state of the chip. Hence, the value stored in those key-gates/registers is expected to remain constant irrespective to the other circuit input variables. Thus, the attacker who does not have prior knowledge about the netlist, can, in principle, differentiate key-gates/registers from other data registers by comparing the EOFM activity of the registers during secure bootup, while applying different sets of inputs.

Another challenge is to differentiate between the combinatorial and sequential logic during EOFM measurement. The clock frequency determined in Sect. IV-B can solve this problem. The adversary can set the chip in free-running mode and scan the whole chip while running the EOFM at clock frequency, as the frequency of interest. This will reveal the clock tree and sequential logic distribution over the entire chip.

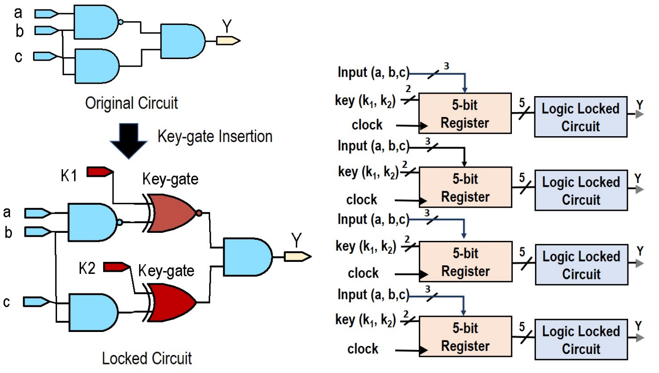
D. Extracting Key Values

The adversary can extract the content of key-registers/gates using the methodology shown in Fig. 3 and described in Sect. IV-C. From analyzing the EOFM activity mapping, the malicious entity can define the value stored in the key-register. If a chip shows activity with the continuous reset loop i.e., a change in stored value from 0 to 1 during the EOFM measurement, it appears as a bright node in the EOFM image. Otherwise, the key-registers storing 0 value appear as inactive register in the EOFM image although the clock tree shows that those registers are active. As a result, the attacker can determine the value stored in each key-register. Once the key bits are exposed, the attack is considered as successful since the functionality of the IP can be unlocked by activating key-gates with necessary inputs. Therefore, the security of the locked IP is no longer impeccable.

V. Experimental Setup

A. Device Under Test

We chose Avalanche FPGA development board designed by Future Electronics as the target platform. It contains a Flash-based Microsemi MPF300 Polarfire FPGA manufactured with 28 nm technology in a flip-chip Ball Grid Array (BGA) package. In this type of package, the silicon die is inverted and placed frontside down. There is no heat sink on top of the package, and hence, we have direct access to the silicon substrate on the backside of the chip without any package preparation or silicon polishing. According to our measurements, the thickness of the substrate is about 700 μm. An 1.3 μm light source is used for acquiring the image of the die without any substrate thinning, see Fig. 5b.



(a) (b)

Fig. 4: (a) Logic locked circuit, where a, b, and c are the inputs and K_1 and K_2 are the key values; (b) Block diagram of implemented circuit where logic locked circuit in Fig.(a) is implemented between 5-bit register block for input signals and key-register.

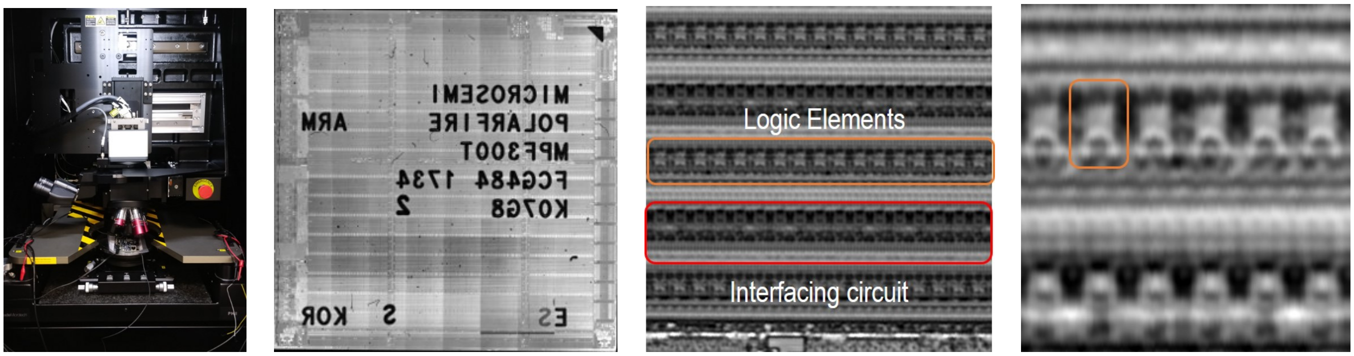
B. Circuit Implementation

For our experiments, we implemented two locked circuits on the Microsemi Polarfire FPGA. The first circuit is a PoC implementation shown in Fig. 4a. For the second and more realistic experiment, we implemented a standard benchmark circuit, namely the benchmark circuit c1355-CS320 [25] which is available at Trust-Hub.org [39].

In case of PoC implementation, the circuit is obfuscated with the XOR/XNOR gates connected with K_1 and K_2 inputs, see Fig. 4a. Once the correct input combination ($K_1 = 1$ and $K_2 = 0$) is applied, the circuit produces the correct output Y. Here, a, b, and c stand for the inputs of the circuit. We implemented four of the logic locked circuit shown in Fig. 4a in a circuit block as shown in Fig. 4b, each logic locked circuit connected to three input registers implemented in parallel representing a, b, and c port in Fig. 4a. The key in our design is 8-bit length as each logic locked circuit in Fig. 4a has two key bits. We also used 8-bit of parallel registers to feed the key to key-gates. In the design, a reset signal is implemented to imitate the reset process in the chip. In real scenario, the attacker can connect a signal generator to the power pins of the chip and reboot the chip to induce the desired frequency for performing EOFM.

C. Measurement Setup

The optical contactless probing setup is provided by a Hamamatsu PHEMOS-1000 FA microscope, see Fig. 5a. The equipment consists of a suitable probing light source (Hamamatsu C13193) and an optical probing preamplifier (Hamamatsu C12323). The development board is placed inside the PHEMOS and a PC is connected to the board to program the FPGA. Programming of the FPGA is performed through USB which is handled by an FTDI chip. The board is powered



(a) (b) (c) (d)

Fig. 5: (a) Optical Probing setup in Hamamatsu PHEMOS-1000 FA microscope for probing Avalanche FPGA development board (b) Reflected light overview image of the complete Microsemi MPF300 Polarfire FPGA die; (c) Zoomed-in view of Fig. 5b showing repeating FPGA logic fabric structure; (c) Zoomed-in view of logic elements (orange rectangle area).

by the provided development board supply. No other electrical modifications were performed on the board.

The setup uses a spectrum analyzer for EOFM analysis. Three objective lenses were used during this work: 5x/0.14 NA, 20x/0.4 NA, 50x/0.76 NA. The 50x lens is equipped with a correction ring for silicon substrate thickness.

VI. Result and Analysis

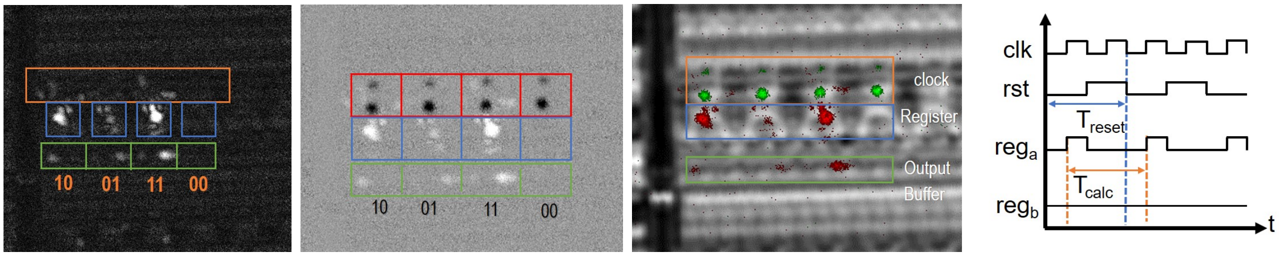
This section presents the results achieved by applying the approach presented in Sect. IV for exposing the key-registers/gates for probing the locking keys. As our logic obfuscated circuitry is implemented in a Microsemi Polarfire FPGA, first, we review briefly the internal structure of this FPGA as a part of the reverse engineering. Afterward, we deploy EOFM for key localization and recognition, respectively (see Sect. IV-C). Finally, we present the results for obfuscation benchmark c1355-CS320 [39]. In the EOFM measurement images, the white and black spots represent the activity of logic elements for two different frequencies. Overlay images are the diffused images of EOFM measurement and reflected light images from the chip. To localize the keys, we have compared two input vectors, $\mathbf{x_0}$ and $\mathbf{x_1}$, where $\mathbf{x_0}$ represents the condition where all the inputs are set to ’0’ and $\mathbf{x_1}$ represents the condition where all the inputs are set to ’1’. Since during bootup process, the chip does not perform any functions, it can be assumed that all the input port are set to inactive or grounded state. Hence, the input vector $\mathbf{x_0}$ can be a representation of the bootup condition of the chip.

A. Profiling Microsemi FPGA

Fig. 5b shows the reflected light overview images of the die acquired with 1.3 μm wavelength. This image is the mirrored panorama image of 9×6 matrix collected with 5x/0.14 NA lens. In the image, the die markings are visible as the chip was not polished. Fig. 5c presents the FPGA logic fabric consists of several identical configurable logic blocks (CLBs). In this figure, the reflected light image is captured with a 50x/0.76 NA. The FPGA logic resources are fabricated as logic clusters as presented in the orange rectangular box in Fig. 5c. The interfacing circuit, which is responsible for the routing between CLBs of the FPGA, is shown in the red rectangle in Fig. 5c. Each cluster consists of twelve logic elements. The rectangular orange box in Fig. 5d shows the further 4x optical zoom-in view of two logic elements in Microsemi FPGA. Each logic element consists of a 4-input LUT with a D-flip-flop. The logic element is fracturable, which means the LUT and flip-flop can be used either together or independently [40]. To map the logical locations of the implementation to the physical one inside the FPGA, we have implemented an 8-bit parallel output registers, connected with an 8-bit key.

The documentation of the development board reveals that the frequency of the internal clock implemented in the board is 50 MHz. Thereafter, using the spectrum analyzer available with the PHEMOS, a clock frequency of 50.14 MHz is defined as the precise clock frequency of the chip. The reset frequency of the chip is set at half the clock frequency, i.e., 25.07 MHz. The EOFM activity is measured and the activity is mapped with the key value applied to the output registers. The activity mapping of output registers are shown in Fig. 6a - 6c. The bright spots marked with blue rectangles in Fig. 6a serve as the output flip-flops of the circuit. The less bright spots confined with orange and green rectangles exhibits the input and output buffer activities of the output register, respectively. The clock and register EOFM activity are subtracted from each other in Fig. 6b. The black and white dots in Fig. 6b represent the clock and register activity, respectively. The red, blue and green rectangles correspond to the clock, register and output buffer activity in the subtracted image. The Overlay image of clock and register activity over the reflected light image is demonstrated in Fig. 6c. The overlay image confirms that each blue rectangle contains two logic elements. The values stored in the flip-flops are shown below the output buffers in Figs. 6a and 6b. In both Figs. 6a and 6b, the rightmost registers does not show any trigger activity for both flip-flops and the output buffer, as the “00” is stored (see Sec. IV-D). The register next to the right most register shows two bright dots at output buffer which implies the stored value is “11”.

The EOFM activity of the register can be explained with the waveform shown in Fig. 6d. In Fig. 6d, two registers, $\mathrm{reg_a}$ and $\mathrm{reg_b}$ are receiving a bit ’1’ and a bit ’0’, respectively. The reset signal is depicted with the waveform rst. $\mathrm{reg_a}$ starts at the logic level low and then changes its state, as soon as the



(a) (b) (c) (d)

Fig. 6: (a) EOFM activity of the register at reset frequency. The stored key value in each register mentioned at the bottom of the corresponding register; (b) Subtracted image of clock activity from register activity, the black and white dots correspond to clock and registers activity, respectively. The stored value in each register mentioned at the bottom of the corresponding register; (c) Overlay image of clock and register activity on reflected light image, where green dots represent clock and red dots represents register activity; (d) Waveforms of the clock (clk), reset signal (rst) and two registers (reg_a and reg_b). The register reg_a receives a signal of bit '1' and the register reg_b receives a signal of bit '0'.

time needed for the preceding calculation (T_{calc}) has elapsed. The rst signal resets the registers with a time period of T_{reset}. As the time period for each consecutive power-on is constant, the time period for T_{calc} is equal to the time period of reset signal, T_{reset}. Therefore, in the EOFM measurement, the reg_a will show its activity. The other register, i.e., reg_b, is carrying a bit '0'. Hence, it will not change its value with the reset signal, and therefore, will not show any activity in the EOFM measurement.

B. Key Extraction from PoC Circuit Implementation

1) Adversary without Access to the Layout: In this subsection, we present how an adversary without access to the GDSII or physical layout information can apply the approach showed in Fig. 3 to reveal the key-register location.

Extracting Clock Distribution: First, to uncover the location of sequential logic, adversary requires to find the clock tree distribution in the chip. Hence, EOFM activity mapping at a clock frequency for two different input vectors, $\mathbf{x_0}$ and $\mathbf{x_1}$, is shown in both Fig. 7a and 8a, respectively. The resulting bright nodes evident in that figure reveals the clock tree distribution and sequential logic elements over the chip. The number of active flip-flops can be identified from the brightness shown in EOFM clock activity (Fig. 7a and Fig. 8a). Hence, by comparing different node intensity in the clock tree EOFM measurements, it has been identified that only one flip-flop is active at the locations marked with blue rectangles in both Fig. 7a and Fig. 8a.

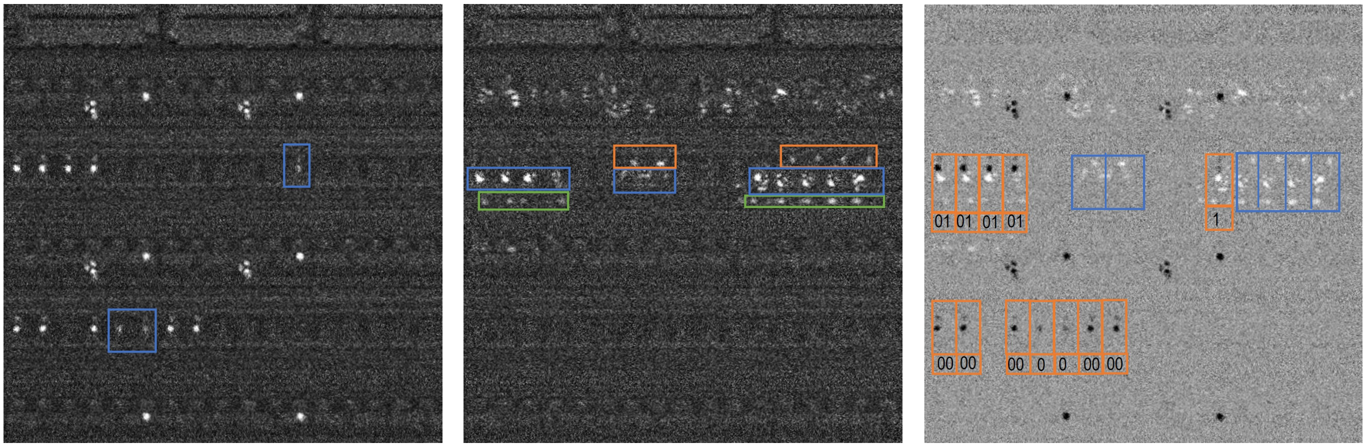
Detecting the Key-registers: Once the EOFM for clock tree is identified, the attacker can apply the approach described in Sect. IV-C to uncover the activity of key-register for different input vectors. The EOFM measurements for both $\mathbf{x_0}$ and $\mathbf{x_1}$ in Fig. 7b and Fig. 8b are collected by rebooting the chip in a continuous loop. These measurements contain the activity of both sequential and combinatorial logic elements. In Fig. 7b and Fig. 8b, the blue, green and orange rectangles represents the logic elements (both sequential and combinatorial), input buffers and output buffers, respectively. To expose the location of sequential elements and registers, an attacker can subtract the EOFM measurement for resetting frequency from clock frequency in PHEMOS, as shown Fig. 7c and Fig. 8c. In both of the images, the black and white dots represent the elements active at the clock and reset frequency respectively. In both images, the orange and blue rectangles represent the registers and combinatorial logic elements, respectively.

Extracting the Key: Applying the method described in Sect. IV-D, the stored value in a register can be identified. Hence, depending on the presence of white spot at the output buffer location of the flip-flops, the stored values are defined in Fig. 7c and Fig. 8c. By comparing the values in aforementioned figures, an adversary without having access to the IP/chip layout can identify nine flip-flops that are maintaining constant output for the different input signals. Thereafter, the chip is operated at free-running mode and EOFM measurement is collected. Eight flip-flops marked with green rectangle in Fig. 8c shows activity in the free-running mode. Therefore the remaining one flip-flop (marked with red rectangle) is identified as the register responsible for continuous resetting of the circuit. Similarly, comparing the EOFM activity and output buffer values of combinatorial logic elements (blue and green rectangles in Fig. 7c and Fig. 8c), the gates implemented in yellow and red rectangle locations can be identified as key-gates and output gates, respectively. Eventually, The key bits are also exposed from EOFM activity. The key bits for this circuit block is 01010101.

2) Adversary with Access to the Layout: The adversary with access to circuit layout can detect the location of key-gates/registers by reverse engineering the layout. Once the key-registers/gates are localized, extracting key value requires only measuring the EOFM activity of those registers/gates. Therefore, she can measure the EOFM activity during secure bootup of the chip and extract the key value in a reset loop as shown in Fig. 7.

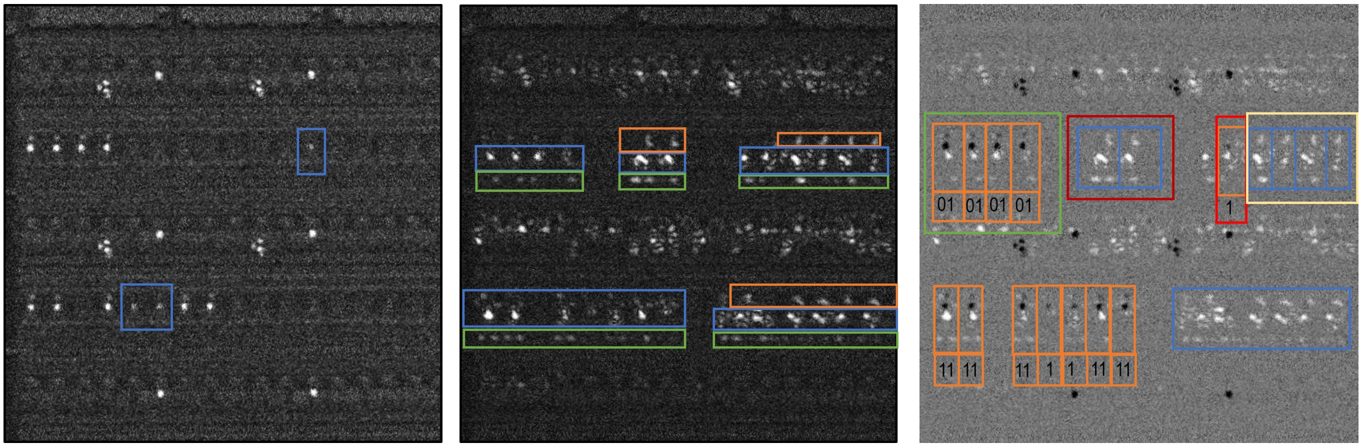
C. Key Extraction from Obfuscation Benchmark Circuit

The methodology described in Sect. IV is used to extract the key used to lock the obfuscation benchmark c1355-CS320. The benchmark is implemented on Microsemi Polarfire FPGA. The benchmark has 41-bit input, 596 gates and locked with a 32-bit key. The EOFM activity of the chip is measured for two input patterns, i.e., $\mathbf{x_0}$ and $\mathbf{x_1}$. The EOFM activity is measured with 50x/0.74NA lens. The measurement for the EOFM activ-



(a) (b) (c)

Fig. 7: Localizing clock and register activity for input vector $\mathbf{x_0}$, (a) EOFM measurement at clock frequency for exposing clock distribution; (b) EOFM measurement at reset frequency for exposing register and combinatorial logic activity; (c) Subtraction image of EOFM activity at clock frequency from EOFM activity at reset frequency where, black and white dots correspond to clock and logic element activity, respectively. The value stored in each register is mentioned at the bottom of the corresponding register.



(a) (b) (c)

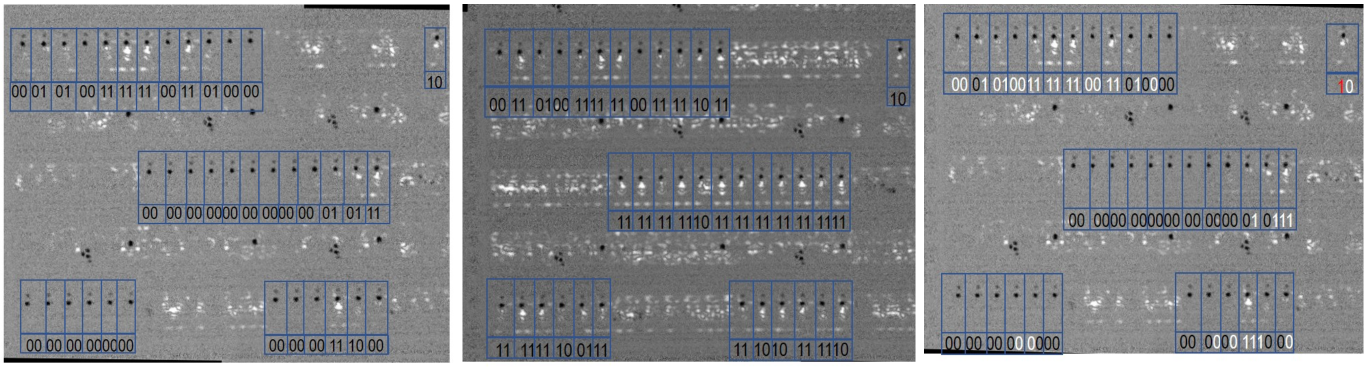
Fig. 8: Localizing clock and register activity for input vector $\mathbf{x_1}$, (a) EOFM measurement at clock frequency for exposing clock distribution; (b) EOFM measurement at reset frequency for exposing register and combinatorial logic activity; (c) Subtraction image of EOFM activity at clock frequency from EOFM activity of reset frequency where, black and white dots represents clock and logic element activity, respectively. The value stored in each register is mentioned at the bottom of the corresponding register. The register in the green box represents the key-register location and red box represents the register responsible for the reset signal.

ity of the complete circuit on the chip is covered in 2×2 matrix and stitched later. The chip is triggered at clock frequency in a loop to expose the clock tree distribution. Thereafter, the chip is triggered in a reset loop to reveal the register activity. Fig. 9a and Fig. 9b represent the subtracted image of EOFM activity at clock frequency from reset frequency for above-mentioned input patterns. From the clock EOFM activity the location of the registers are identified and showed in blue rectangles in Fig. 9a and Fig. 9b. The values stored in all the registers are probed (the stored data are presented in Fig. 9a and Fig. 9b). As discussed in Sect. IV, the register maintaining the same state irrespective to applied input patterns are identified as the key-registers. We identified the registers storing values written in white color in Fig. 9c are maintaining constant state irrespective to the input vector. Therefore, the key value for the logic locked benchmark circuit is identified. The register with red colored value in Fig. 9c, is the reset signal. An adversary with complete reverse engineering capability can smoothly localize the key-registers and directly probe those registers to uncover the secret keys.

VII. Discussion

A. Challenges for an Adversary

We demonstrated that the key extraction from an obfuscated circuit is feasible. However, there might be a few challenges in a real scenario attack for the adversary.



(a) (b) (c)

Fig. 9: Localizing clock and register activity for input pattern $\mathbf{x_0}$ and $\mathbf{x_1}$ condition in obfuscation benchmark c1355-CS320. Here, for all the images, black and white dots represents clock and logic elements activity, respectively. The stored data in each register for input $\mathbf{x_0}$ and $\mathbf{x_1}$ is mentioned at the bottom of the corresponding register. (a) Subtracted image of EOFM measurement for clock and reset frequency for $\mathbf{x_0}$.; (b) Subtracted image of EOFM measurement for clock and reset frequency for $\mathbf{x_1}$; (c) Detecting the key locations and extracting keys from EOFM activity showed in Fig. 9a and Fig. 9b. The stored data with white color represents the location and value stored in the key-register and the stored data with red color represents the register connected to reset signal.

Reset Counter in the Device: Optical probing requires several repeated measurements. In modern processors and FPGAs, the number of reboot attempts on a chip can be monitored by setting up a counter. Hence, If the number of rebooting attempts exceeds the predefined threshold, the key values saved in the tamper-proof memory can be zeroized [28]. However, an attacker with access to the layout can detect such counter and remove it from the netlist through FIB circuit edit. Especially, if the adversary is an untrusted foundry, such an edit is conceivable.

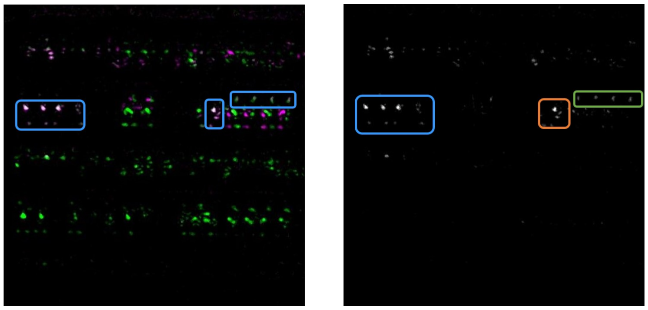
Extracting the Exact Key Sequence: An adversary without having access to the locked circuit layout might face challenge to determine the right order for the exact key bits, since she does not have any knowledge about the order of the registers. However, the IP functionality and implementation is still compromised, as described in Sect. IV-D, since the locking key bits are exposed.

Cost and Time Required for Key Extraction: An adversary without access to the layout or the gate-level netlist can localize key-gates/registers using optical probing. The optical setup used in the paper is a common FA tool, which can be rented for about \$300/h, including operator from different labs. The time spent on the microscope to localize the registers for an end-user without having the layout does not exceed 8 days (8h/d). Thus, the cost would be less than \$20k.

On the other hand, the complete reverse engineering of the chip requires access to delayering and imaging tools like plasma etcher and SEM to acquire the image of metal layers, vias, polysilicon and active layers of the chip. All these equipment are also accessible in many academic/industry labs and can be rented for only a few hundreds of dollars per hour. Thereafter, the layout and netlist of the chip can be extracted using reverse engineering software like Pix2Net [41]. In common belief, reverse engineering is a time and labor intensive task. However, automation and advancement in FA tools and automated netlist extraction tools have turnaround that belief. Besides, once the netlist extraction of the chip is completed, the design and locking key of all chips from the same family is available to the adversary. Hence, complete reverse engineering and probing the locking key can be compared with ”attack one, attack all” approach.

Scalability of the Attack: Localizing the key-registers/gates requires identifying the common operating node irrespective to input applied to the chip/IP. With billions of transistors integrated into a single chip, the challenge of localizing the common activity nodes can be addressed by image processing and computer vision techniques as well [30]. Applying simple image registration (see Fig. 10a), subtraction (see 10b), or image correlation, in Fig. 7b and Fig. 8b, can localize the common activity nodes in the chip. The only obstacle is, an adversary may need to increase the number of input patterns used for collecting EOFM measurements. Moreover in Sect. VI-C we have shown how we extracted a 32-bit key from a 596 gated, c1355-CS320 obfuscation benchmark using optical probing. However, it is essential to mention that our approach is scalable since increasing the size of the key input would linearly increase the laser scanning time.

Localizing the Key-gates: The key can be directly fed to the key-gates from key storage by eliminating the key-registers. This can significantly increase the difficulty in localizing the key-gates for an adversary without access to the layout. Since the key has to be available at the key-gates after each power-on (See the blue rectangles in Fig. 7c), an adversary with full-blown reverse engineering capability can localize the key gates and probe the key value (see the yellow rectangle in Fig. 8c). Therefore, feeding the key directly to combinational logic does not eliminate the threat imposed by the probing attack against logic locking. In addition, directly feeding the key to the key-gates means connecting the key storage to the gates



(a) (b)

Fig. 10: Comparing EOFM activity of $\mathbf{x_0}$ and $\mathbf{x_1}$ for Localizing key-registers/gates through image processing using (a) image registration; (b) image subtraction. In both figures rectangle areas shows common activity nodes (key-registers and key-gates) for both Fig. 7b and Fig. 8b.

through interconnects. This significantly increase the threat of electrical probing, since localizing the key-storage is sufficient for an adversary with access to the advanced FA tools.

B. Applicability of The Attack

A malicious entity can apply the proposed methodology for all the logic obfuscation techniques which use key vectors to lock the design and functionality. This methodology is equally applicable for sequential obfuscation methods like finite state machine (FSM) based obfuscation for IP protection. In the case of FSM-based obfuscation, the FSM offers two distinct modes of operation for the IP core, i.e., normal and obfuscated [7]. The operation mode of the IP relies on the applied key value. Similar to combinational logic locking, FSM-based obfuscation also assumes that the keys are stored in a tamper-proof/secure memory. Consequently, an adversary can break into a locking scheme by reading out the key from the key-registers/gates. Similarly, in other logic locking schemes, such as the Stripped-functionality logic locking (SFLL) approach [16], which is the current state-of-the-art countermeasure against oracle-guided attacks, the key cannot be protected by the assumption of *tamper-proof* and *read-proof* memory.

VIII. Potential Countermeasures

The success of the proposed attack in this paper depends mainly on two steps, namely accessing the chip from the backside and localizing the registers. For this reason, to safeguard the confidentiality of the IP, logic locking requires both detection and prevention of unauthorized access into the chip. Possible countermeasures can be integrated into the chip during packaging, device fabrication, and circuit design.

Package Level Protection: The proposed attack reveals the key value from silicon backside, and the continuous increase of interconnect layers at the frontside of the chip demands a secured chip backside. At the packaging level, such protection can be provided by adding active opaque layers to the backside of the chip. As such layers can easily be removed, an active monitoring scheme must be implemented to detect an adversarial attack.

Device Level Protection: At the device level, optical probing sensors can be deployed to detect an attack attempt. Since the optical beam stimulates the active regions thermally, conventional photosensors fail to trigger during optical probing. Nonetheless, the thermal stimulation introduces temperature and current variations in the circuit, which can influence circuits, such as ring-oscillators (ROs) [42]. In this case, implementing ROs as a probing protection scheme can generate an anti-tamper reaction in the chip to protect the locking keys. In [43] nanopyramid structures are implemented in selective areas inside the chip to mitigate optical probing attacks by scattering the reflected laser beam, and consequently, scrambling the measurements of the register contents.

Circuit Level Protection: Physical attack methods, such as optical attacks and microprobing, rely on the electrical test and structural characterization to detect a region of interest. Thus, a circuit-level solution can be widely accepted for the semiconductor industry. As the logic locking key is static and embedded in the device memory, it can be probed by the aforementioned attacks. Hence, randomizing the sate of the key-register during the boot-up of locked IP can be a solution against optical probing. A true random number generator (TRNG) can be used to store random values at the key-register. Nonetheless, storing random value in the key-register may introduce reliability issues for the chip. Therefore, additional control circuitry is require to store the random values during the inactive state of the corresponding IPs. However, integrating TRNG in the circuit may introduce additional area and delay overhead to the chip. In addition, untrusted foundry can localize and remove the TRNG through reverse engineering and circuit edit, respectively [30], [44]. Another solution would be the usage of dummy active registers connected to functional gates to disguise the key-registers and eventually hiding the key-gates. However, the circuit level countermeasures might be known to a malicious foundry, and they can be easily deactivated. However, it still can be considered more secure against end-users.

IX. Conclusion

In this work, we presented that irrespective of the security of the locking schemes, storing the key on the same chip makes the entire obfuscation vulnerable to adversaries with different capabilities. Unfortunately, to this date, researchers have focused on securing the IP by inserting more gates, sacrificing area and power overhead, believing that the key is safe under the roof of tamper/read-proof memories. In other words, we demonstrated that even if tamper-proof or secure memories exist, the key movement between the memory and key gates of the locked circuit during the bootup process of a chip creates a side-channel leakage, which can be used by an attacker to extract the key. We further evaluated the capabilities of different classes of adversaries with or without access to the chip layout. Based on the capabilities of adversaries, we showed how an attacker in each class of adversaries could deploy FA tools to read out the key. To validate our claims, we mounted an optical probing attack against a proof-of-concept locked circuit as well as a standard obfuscation benchmark, implemented on a 28 nm flash-based FPGA, and successfully extracted the key. We discussed the challenges that an adversary might face in a real-scenario attack, and how the proposed attack technique can be applied to other locking schemes, such as FSM-based techniques. Finally, we proposed potential countermeasures, which makes the key extraction more challenging.

REFERENCES

[1] B. Shakya, M. M. Tehranipoor, S. Bhunia, and D. Forte, "Introduction to hardware obfuscation: Motivation, methods and evaluation," in *Hardware Protection through Obfuscation*. Springer, 2017, pp. 3–32.

[2] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[3] Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security." in *USENIX security symposium*, 2007, pp. 291–306.

[4] "Ieee 1735-2014 - ieee recommended practice for encryption and management of electronic design intellectual property (ip)," https://standards.ieee.org/standard/1735-2014.html, accessed: 2018-07-26.

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

[6] R. S. Chakraborty and S. Bhunia, "Hardware protection and authentication through netlist level obfuscation," in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE Press, 2008, pp. 674–677.

[7] ——, "Harpoon: an obfuscation-based soc design methodology for hardware protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.

[8] G. K. Contreras, M. T. Rahman, and M. Tehranipoor, "Secure split-test for preventing ic piracy by untrusted foundry and assembly," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 196–203.

[9] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 709–720.

[10] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 137–143.

[11] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "Appsat: Approximately deobfuscating integrated circuits," in *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 95–100.

[12] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Security analysis of anti-sat," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 2017, pp. 342–347.

[13] 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 *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 189–210.

[14] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 83–89.

[15] Y. Xie and A. Srivastava, "Anti-sat: Mitigating sat attack on logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[16] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. J. Rajendran, and O. Sinanoglu, "Provably-secure logic locking: From theory to practice," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1601–1618.

[17] M. Hoffmann and C. Paar, "Stealthy opaque predicates in hardware-obfuscating constant expressions at negligible overhead," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 2, pp. 277–297, 2018.

[18] " Techinsight Inc." [Online]. Available: https://www.techinsights.com

[19] H. Lohrke, S. Tajik, C. Boit, and J.-P. Seifert, "No place to hide: Contactless probing of secret data on fpgas," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 147–167.

[20] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit, "On the power of optical contactless probing: Attacking bitstream encryption of fpgas," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1661–1674.

[21] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert, "Breaking and entering through the silicon," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 733–744.

[22] R. Anderson and M. Kuhn, "Tamper resistance-a cautionary note," in *Proceedings of the second Usenix workshop on electronic commerce*, vol. 2, 1996, pp. 1–11.

[23] J. Rajendran, H. Zhang, C. Zhang, G. S. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault analysis-based logic encryption," *IEEE Transactions on computers*, vol. 64, no. 2, pp. 410–424, 2015.

[24] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing ic piracy using reconfigurable logic barriers," *IEEE Design & Test of Computers*, vol. 27, no. 1, 2010.

[25] S. Amir, B. Shakya, X. Xu, Y. Jin, S. Bhunia, M. Tehranipoor, and D. Forte, "Development and evaluation of hardware obfuscation benchmarks," *Journal of Hardware and Systems Security*, pp. 1–20, 2018.

[26] F. Courbon, S. Skorobogatov, and C. Woods, "Reverse engineering flash eeprom memories using scanning electron microscopy," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2016, pp. 57–72.

[27] M. T. Rahman, M. S. Rahman, H. Wang, S. Tajik, W. Khalil, F. Farahmandi, D. Forte, N. Asadizanjani, and M. Tehranipoor, "Defense-in-depth: A recipe for logic locking to prevail," *arXiv preprint arXiv:1907.08863*, 2019.

[28] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert, "Key extraction using thermal laser stimulation," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 573–595, 2018.

[29] H. Photonics, "Emission Microscopy: Phemos-1000," https://www.hamamatsu.com/resources/pdf/sys/SSMS0003E_PHEMOS1000.pdf, accessed:2018-04-26.

[30] M. T. Rahman, Q. Shi, S. Tajik, H. Shen, D. L. Woodard, M. Tehranipoor, and N. Asadizanjani, "Physical inspection & attacks: New frontier in hardware security," in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*. IEEE, 2018, pp. 93–102.

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

[32] L. Bossuet, M. Grand, L. Gaspar, V. Fischer, and G. Gogniat, "Architectures of flexible symmetric key crypto engines—a survey: From hardware coprocessor to multi-crypto-processor system on chip," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 41, 2013.

[33] N. Asadizanjani, M. Tehranipoor, and D. Forte, "Pcb reverse engineering using nondestructive x-ray tomography and advanced image processing," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 2, pp. 292–299, 2017.

[34] B. R. Beachem and M. K. Smith, "Key management to protect encrypted data of an endpoint computing device," Nov. 19 2013, uS Patent 8,588,422.

[35] B. V. Patel, "Method for securing communications in a pre-boot environment," Dec. 4 2001, uS Patent 6,327,660.

[36] N. S. N.V., "Realizing Today's Security Requirements: Achieving End-To-End Security with a Crossover Processor," https://www.nxp.com/docs/en/white-paper/IMXRTCROSSWP.pdf, NXP, Tech. Rep., 8 2017, accessed: 2018-09-30.

[37] A. Mundra and H. Guan, "Secure boot on embedded sitara$^{\text{TM}}$ processors," http://www.ti.com/lit/wp/spry305a/spry305a.pdf, accessed: 2018-09-30.

[38] M. Werner, R. Schilling, T. Unterluggauer, and S. Mangard, "Protecting risc-v processors against physical attacks," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1136–1141.

[39] "Trust-hub benchmark c1355-cs320," http://www.pld.ttu.ee/~maksim/benchmarks/iscas85/verilog/.

[40] M. Corporation, "User guide: Polarfire fpga fabric," https://www.microsemi.com/document-portal/doc_view/136522-ug0680-polarfire-fpga-fabric-user-guide, accessed: 2018-07-14.

[41] " MicroNet Solutions, Inc." [Online]. Available: http://micronetsol.net/pix2net-software/

[42] S. Tajik, J. Fietkau, H. Lohrke, J.-P. Seifert, and C. Boit, "Pufmon: Security monitoring of fpgas using physically unclonable functions," in

On-Line Testing and Robust System Design (IOLTS), 2017 IEEE 23rd International Symposium on. IEEE, 2017, pp. 186–191.

[43] H. Shen, N. Asadizanjani, M. Tehranipoor, and D. Forte, “Nanopyramid: An optical scrambler against backside probing attacks,” in *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*. ASM International, 2018, p. 280.

[44] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, “Probing attacks on integrated circuits: Challenges and research opportunities,” *IEEE Design & Test*, vol. 34, no. 5, pp. 63–71, 2017.