The Key is Left under the Mat

On the Inappropriate Security Assumption of Logic Locking Schemes

Mir 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 Intellectual Property (IP) piracy by untrusted entities in the design and fabrication process. It obfuscates the netlist by adding extra key-gates, to mislead an adversary, whose aim is to reverse engineer the netlist.The correct functionality will be obtained only if a correct key is applied to the key-gates. The key is written into a nonvolatile memory (NVM) 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 researches 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, no matter how robust is the logic-locking scheme. 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 common logic locking benchmarks on a 28 nm Flash-based FPGA and extract their keys using optical probing. Finally, we discuss the feasibility of the proposed attack in different scenarios and propose potential countermeasures.

Keywords: Logic Locking, Optical Contactless Probing, Tamper-proof Memory

1 Introduction

The supply chain of integrated circuits (ICs) has changed significantly over the past two decades. As the advancement in the semiconductor industry has continued to provide smaller technology nodes, more complex and sophisticated fabrication facilities are required to achieve the economic scale of production. Besides, due to the high demand for reduction in manufacturing cost and shortened time-to-market, the globalization of semiconductor manufacturing has been on the rise. Hence, the business model for the semiconductor industry has shifted from the vertical model towards the horizontal model (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. The success of the horizontal business model became evident when fabless companies like Qualcomm and Nvidia broke into the top twenty semiconductor industry [BTZ10]. However, with

many entities involved in design, manufacturing, integration, and distribution that are located across the globe, original IP owners no longer have control over the entire supply chain [STBF17, GDT14]. Hence, 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 above-mentioned protection schemes, researchers have focused on developing active approaches like IC metering [AK07], IP encryption [IEE], logic locking/ obfuscation [RKM10, CB08], state space obfuscation [CB09], secure split manufacturing [CRT13, JM07], and IC camouflaging [RSSK13].

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 IP owner through a nonvolatile memory (NVM) once 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 [SRM15, SLM+17], SPS attacks [YMSR17], bypass attacks [XSTF17], and key sensitization attacks [RPSK12]. These attacks are mostly dependent on analysis of input/output patterns received from an unlocked chip, and hence, they are referred to as Oracle-guided attacks.

Over the past several years security community has focused on developing the countermeasure to hinder those Oracle-guided attacks [XS18, YSN+17, HP18]. While protection against the above-mentioned attacks received so much attention, 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. As a result, the only way to break the obfuscation is to develop an Oracle-guided attack to deobfuscate the netlist, and consequently, obtain the correct functionality of the circuit. 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.

An adversary, such as untrusted foundry, is equipped with the most advanced failure analysis (FA) equipment. FA equipment, such as scanning electron microscopes (SEM) or laser scanning microscopes (LSM), are powerful tools, which can assist to extract data from a chip [RST+18]. Hence, it is conceivable that an untrusted foundry gets access to the shipped product in the market and use their capabilities to extract the unlocking key from the obfuscated chips. Techniques, such as optical probing [LTBS16, TLSB17] or microprobing [HNT+13, AK96], can be employed to localize points of interests and probe the key movement between the memory and the locked logic. Besides, the aforementioned assumptions do not consider the threat imposed by an end user through full-blown or partial reverse engineering. Recent advancements in reverse engineering toolsets, imaging techniques, automated and non-destructive netlist extraction methods have freed the end user from the constrains imposed by difficulties of reverse engineering. Nonetheless, the partial reverse engineering capability gained from FA tools [RST+18, NSSO12, TNH+14] also raise the omen of ransom by malicious end user. Hence, an in-depth analysis of attack models for logic locking algorithms are required to fill the voids in countermeasures proposed to secure the obfuscated IP.

Our Contribution The primary aim of this work is to evaluate the practicability of extracting key values for unlocking a logically locked circuitry by either untrusted foundries or end users, with or without having access to the circuit layout. We first present the attack models for the complete life cycle of a modern chip. For this purpose, we have

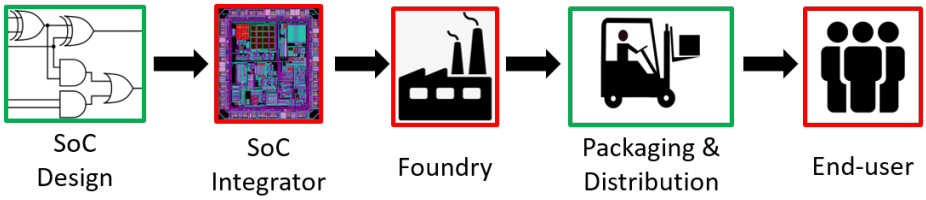


Figure 1: Horizontal model of supply chain in semiconductor industry. In case of logic locking it is assumed the designer, packaging and distribution is trusted (green box) while other phases may not (red box).

analyzed the assets available for different adversaries like System on Chip (SoC) integrator, untrusted foundry, or end user. We further present the challenges for an attacker to fulfill her objectives. We also demonstrate that the assumption of having a tamper- and read-proof memory for the key storage is not sufficient to prevent IP piracy. 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. We demonstrate that an attacker can localize and probe the key-gates or registers using optical probing from the IC backside in a non-destructive way. As a result, the entire key can be extracted. 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. To validate our claims, we conducted experiments on two different logic locking implementations on a Microsemi Polarfire FPGA fabricated in 28 nm node technology. The first implementation consists of a simple logic locked circuit as proof-of-concept implementation, and the second one contains a standard logic obfuscated benchmark circuit. 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. Finally, we propose possible countermeasures to mitigate the shortcomings of the current logic locking schemes.

2 Background

2.1 Logic Locking

Logic locking has been developed as an obfuscation technique to conceal the functionality, design, and layout of IP cores, in order 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 by the correct key set connected to key-gates, Otherwise, the function does not reveal correct input-output behaviour. The key is not available during the fabrication process and is inserted into an NVM (e.g., Flash, EEPROM, e-fuse/ one-time programmable memory) inside the chip before releasing the chip into the market. Consequently, the correct functionality is hidden from an untrusted foundry during manufacturing. Since random insertion of key-gates, without considering circuit structure, does not add significant security features to the design, several key-gate insertion algorithms have been proposed in the literature, like the insertion of XOR/XNOR gates [RZZ+15, RKM10], multiplexers [RZZ+15] and lookup tables [BTZ10].Among different gate insertion methods, XOR/XNOR based logic locking offers higher output corruptibility at lower area, power, and delay overhead. It is also considered as the most acceptable logic element [XS18]. Nonetheless, when obfuscating a circuit with multiple modules, each module is obfuscated with a number of key bits that

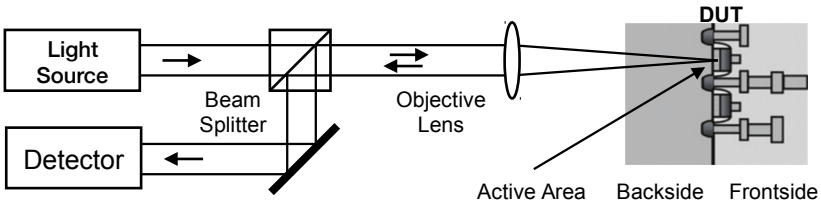


Figure 2: Simplified illustration of contactless optical probing signal acquisition [TLSB17].

are proportional to the size of the module. Hence, a larger module is obfuscated with more locking gates than the smaller ones [ASX+18]. 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.

2.2 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 way 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 [CSW16]. 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 scanning electron microscopy (SEM) [KKF+18], or battery-backed RAMs can be read out by optical techniques, such as thermal laser stimulation (TLS) [LTK+18]. Physically unclonable functions (PUFs) have demonstrated similar vulnerabilities to optical techniques as well [LTBS16].

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. 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 netlist is available to an adversary with significant FA capability (untrusted foundry or a reverse engineering entity) which means these entities should be well equipped to carry a rather straightforward non-destructive attack, such as optical probing, and easily steal the key.

2.3 Optical Contactless Probing

Optical FA techniques have been promoted as a solution for contactless IC debugging from the backside of a chip. As the semiconductor industry approaches even smaller technology nodes and continuous increase in interconnect layers, chip debugging from the front side is becoming more complex. Nonetheless, contactless interaction with the transistors requires much less effort in comparison to other debugging tools, such as Focused Ion Beam (FIB) circuit editing [TLSB17]. Besides, transparency of silicon to photons in near-infrared (NIR) spectrum aligned with the popularity of flip-chip packaging has also shed light

on optical analysis for the operating chip. The optical contactless probing is one such laser-based optical methods, offers IC debugging in a non-destructive and contactless manner [TLSB17]. 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 with NIR wavelengths pass through the backside of silicon substrate which leads to partial absorption and reflection at interfaces such as active region or first metal layer interconnect. In the case of EOP, the electrical signal at a node modulates the amplitude and phase of reflected light. The modulated light is fed to an optical detector to measure the intensity of light as shown in Fig. 2. Since, the modulation of the reflected light signal is small, a sufficient signal to noise ratio is required for probing. This can be achieved by running the signal in a trigger loop and measuring the signal several times.

While EOP focuses on a single transistor, 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 [Pho]. 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 or false color representation [TLSB17]. 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

2.4 Reverse Engineering

Reverse Engineering has several meanings in the context of hardware security. In this work, we make distinction between *full-blown* reverse engineering and *partial* reverse engineering. 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 [RST+18, TJ09, QCF+16]. Through full-blown reverse engineering complete layout, netlist, and functionality extraction 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. Information collected from partial reverse engineering can be used to initialize malicious activities.

3 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. We discuss resources, tools, and expertise possibly available for an attacker, and assess how they can be used by her to perform the attack. Our attack method is motivated by the fact that the logic locking key is stored in a NVM e.g., flash or eFuses. During the bootup of any chip, to speed up the functionality of the chip and avoid latency due to key reading form NVM, the key values are transferred from the memory to key-gates through registers [BGG+13]. 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

Table 1: A comprehensive attack model for logic locking

Attacker	Asset holding	Challenges	Advantages	Objective
Untrusted Foundry	GDSII, Unlocked chip	Reverse engineering layout to localize key-gates/registers	Layout availability	1. IP Piracy 2. Extracting key for ransome
SOC integrator	Soft/hard IP, Unlocked chip	Reverse engineering IP to localize key-gates/registers	1. Netlist availability 2. Knowledge about chip/IP functionality	
End user	Unlocked chip	Unavailability of gate -level netlist, hence need to perform complete or partial reverse engineering on the chip	1. Availability of I/O pin configuration 2. Knowledge about chip functionality	

is to include all possible adversaries in the supply chain. Eventually, available resources, advantages, and challenges for those adversarial entities are analyzed.

3.1 Untrusted Foundry

A foundry has access to state-of-art reverse engineering and failure analysis tools. Besides, it has access to physical layout and GDS II 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 GDS II 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. These activated chips allow access to the inputs, and consequently, the outputs can be observed.

3.2 SoC Integrator

An SoC integrator has access to the hard/soft IP core as well as the functionality of the chip. Besides, she can also get access to the unlocked chip available in the open market. Such availability of layout and functionality could gives the designer a similar motivation like untrusted foundry for IP piracy through unlocking the IC functionality.

3.3 End User

An end user can have temporarily access to reverse engineering capabilities similar to facilities available to a foundry. However, she does not necessarily have access to the layout and GDS II file. However, she can gain knowledge of key-gate/register location through delayering, imaging and netlist extraction by reverse engineering the chip. The motivation for the end user with the aforementioned capabilities is also similar to the untrusted foundry. This is possibly the most inexpensive case and the most dangerous one, where a single person can be the attacker. This kind of adversary is the primary focus of this paper. From documentation available along with access to the unlocked chip, the adversary can learn about the functionality of the chip.

The assets, challenges and advantages available to the untrusted foundry, SoC integrator and end user are summarized in Table 1.

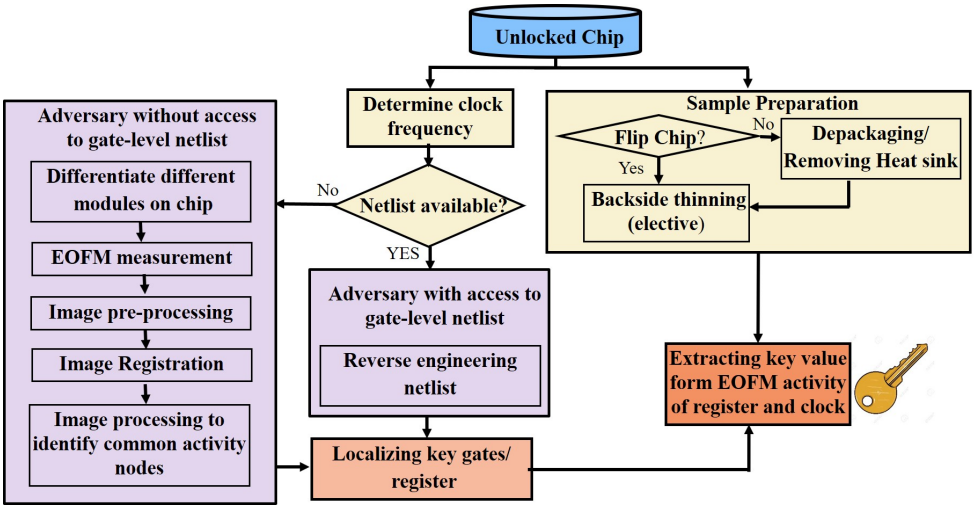


Figure 3: Simplified illustration of key extraction methodology from logic obfuscated circuitry using optical contactless probing.

4 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, SoC integrators, or end users with reverse engineering capabilities, we assume the following information is available. First, the GDS II 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 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.

4.1 Sample Preparation

Earlier non-flip-chip ICs 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 a 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 expose the silicon on the backside of the chip, and thus, the chip is ready for optical analysis. The adversary can deploy X-ray imaging [ATF17] 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.

4.2 Determining Clock Frequency

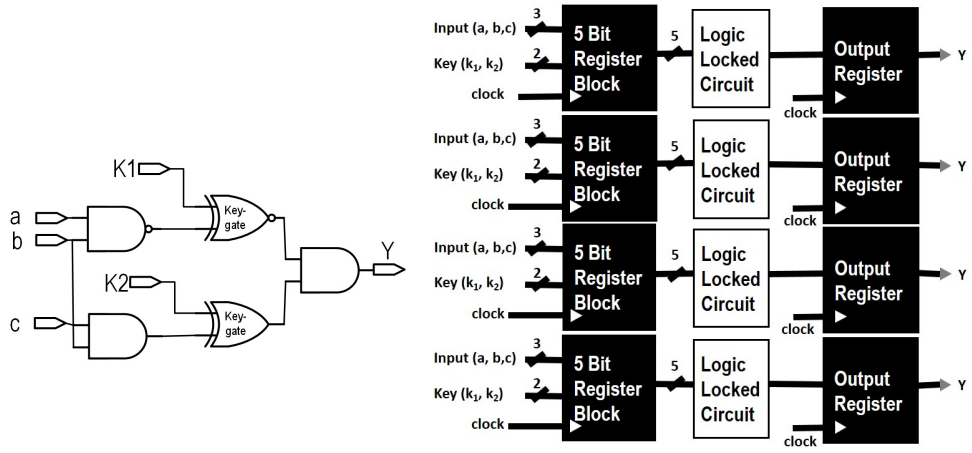
The registers in an IC are connected to the clock tree for a standard chip architecture. Thus, For the end user who does not have knowledge about the location of key-registers, the clock frequency plays the crucial role by revealing the location of sequential logic elements on the chip using EOFM. Determining the clock frequency is an iterative method. For this purpose, the attacker can analyze 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 clock.

4.3 Chip 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 GDS II 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 malicious end user focuses on the logic area as possible location for key-gates and key-registers.

To reveal the exact location of key-register without access to gate-level netlist, 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 secured memory [BS13, Pat01]. These keys are used for authenticating the operating system [N.V17, MG]. 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. 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.



(a) (b)

Figure 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. 4a is implemented between 5 bit register block and output registers.

Another challenge is to differentiate between the combinatorial and sequential logic during EOFM measurement. The clock frequency determined in Sect. 4.2 can solve this problem. She 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 ASIC area. Even an adversary with layout or gate-level netlist can use the EOFM activity of clock to localize the registers and focus on identifying the key-registers.

4.4 Extracting Key Values

The adversary can extract the content of key-registers/gates using the methodology shown in Fig. 3 and described in Sect. 4.3. From the EOFM activity, the malicious entity can define the value of stored data in the register. The value stored in the key-register can be identified by analyzing the activity mapping in EOFM image. If a chip shows activity with the continuous reset loop i.e., change in stored value from 0 to 1 during 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 register is active. As a result, the attacker can determine the value stored in the key-registers.

5 Experimental Setup

This section describes the device under test, circuit implementation, and the measurement setup used to realize the approach discussed in Sect. 4.

5.1 Device Under Test

We chose Avalanche FPGA development board designed by Future Electronics as the target platform. It contains a 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. According to our measurements, the thickness of the substrate is about 700 μm. Silicon is transparent to the 1.3 μm light source, and therefore, an image of the die can be acquired without any substrate thinning, see Fig. 5(a). To conduct an optical attack from the backside of the chip, no package preparation or silicon polishing is required.

5.2 Circuit Implementation

For our experiments, we implemented two locked circuits on the Microsemi Polarfire FPGA. The first circuit is a proof-of-concept (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 [ASX+18] which is available at Trust-Hub.org [Ben].

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. We also used 8-bit of parallel registers to feed the key to key-gates. The complete design is presented as a block diagram in Fig. 4b.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.

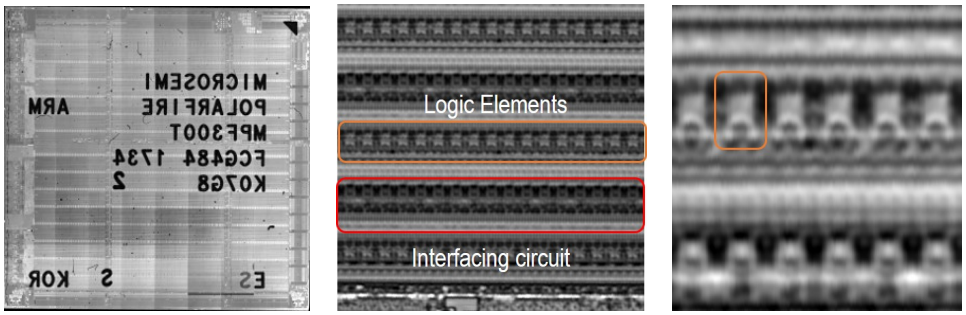
5.3 Measurement Setup

The optical contactless probing setup is provided by a Hamamatsu PHEMOS-1000 failure analysis microscope. 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. When actively being used, the board is powered by the provided development board supply. No other electrical modifications were performed on the board.

The setup uses a spectrum analyzer for EOFM while EOP waveforms are acquired using the PHEMOS software. 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. The optical path is as follows: Photons with a wavelength of 1330 nm are emitted by the light source. The emitted light is deflected by galvanometric mirrors, and then focused through the objective lens into the DUT. The reflected light from the DUT is passed on to a detector, and the detector signal is fed into the preamplifier. The output of the preamplifier can then be fed into the digitizer for averaging and EOP waveform acquisition with a stationary beam. Alternatively, the signal is fed into the spectrum analyzer for generation of EOFM activity maps by scanning the optical beam.

6 Result and Analysis

This section presents the results achieved by applying the approach presented in Sect. 4 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 and image processing for key localization and recognition, respectively (see Sect. 4.3). Finally, we present the results for obfuscation benchmark c1355-CS320 [Ben]. In the EOFM measurement images, the white and black spots represent the activity of logic elements for



(a) (b) (c)

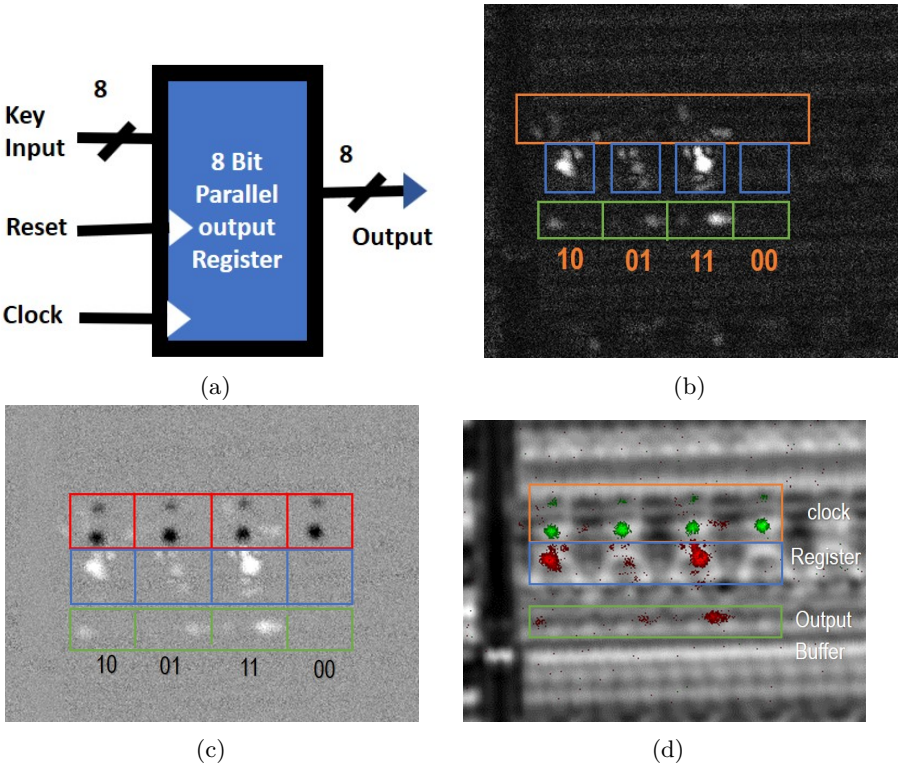
Figure 5: (a) Reflected light overview image of the complete Microsemi MPF300 Polarfire FPGA die; (b) Zoomed-in view of Fig. 5a showing repeating FPGA logic fabric structure; (c) Zoomed-in view of logic elements (orange rectangle area).

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.

6.1 Profiling Microsemi FPGA

Fig. 5a 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. 5b 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, followed by correcting the thickness of the die with the correction ring attached to the lens. The FPGA logic resources are fabricated as logic clusters as presented in the orange rectangular box in Fig. 5b. The interfacing circuit, which is responsible for the routing between CLBs of the FPGA, is shown in the red rectangle in Fig. 5b. Each cluster consists of twelve logic elements. The rectangular orange box in Fig. 5c shows the further 4x optical zoom-in view of logic elements in Microsemi FPGA. The rectangular box in Fig. 5c comprises two logic elements. 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 [Cor]. To map the logical locations of the implementation to the physical one inside the FPGA, we have implemented an 8-bit parallel output registers (Fig. 6a). An 8-bit key is loaded into the 8-bit register simultaneously.

Identifying the sequential element in the design is possible through the clock frequency of the chip. The documentation of the development board reveals that the frequency of the internal clock implemented in the board is 50 MHz. Thereafter, using the iterative method and the spectrum analyzer available with the PHEMOS, a clock frequency of 50.14 MHz is defined as the precise frequency of the chip. The reset frequency of the chip is set at half the clock frequency, i.e., 25.07 MHz. Along with this, the filter bandwidth for conducting EOFM is set at 1 kHz on the spectrum analyzer. 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. 6b - 6d. The bright spots marked with blue rectangles in Fig. 6b 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



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

Figure 6: (a) Block diagram of the circuit implemented in Microsemi MPF300 Polarfire FPGA for logical to spatial profiling; (b) EOFM activity of the register at reset frequency. The stored key value in each register mentioned at the bottom of the corresponding register; (c) 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; (d) Overlay image of clock and register activity on reflected light image, where green dots represent clock and red dots represents register activity.

from each other in Fig. 6c. The black and white dots in Fig. 6c represent the clock and output 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. 6d. 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. 6b and 6c. In both Figs. 6b and 6c, the rightmost registers do not show any activity for both the flip-flops and the output buffer. As the "00" is stored in both of the registers, no trigger activity is visible in that section (see Sec. 4.4). 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. 7. In Fig. 7, two registers, reg_a and reg_b are receiving a bit '1' and a bit '0', respectively. The reset signal is depicted with the waveform rst. reg_a starts at the logic level low and then changes its state, as soon as the time needed for the preceding calculation (T_{calc}) has elapsed. As the reset input goes high, reg_a is reset and afterward, once the reset goes low, the power-on cycle is restarted. 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.

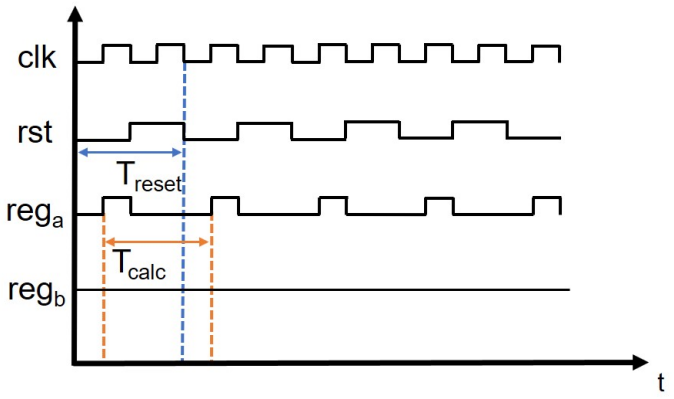
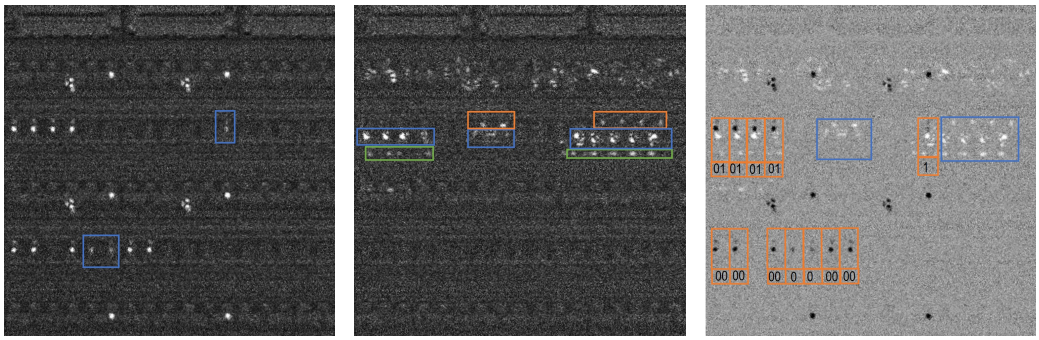


Figure 7: Waveforms of the clock (clk), reset signal (rst) and two registers (reg_a and reg_b). The register reg_a receives a signal of bit '1' and the register reg_b receives a signal of bit '0' .



(a) (b) (c)

Figure 8: 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.

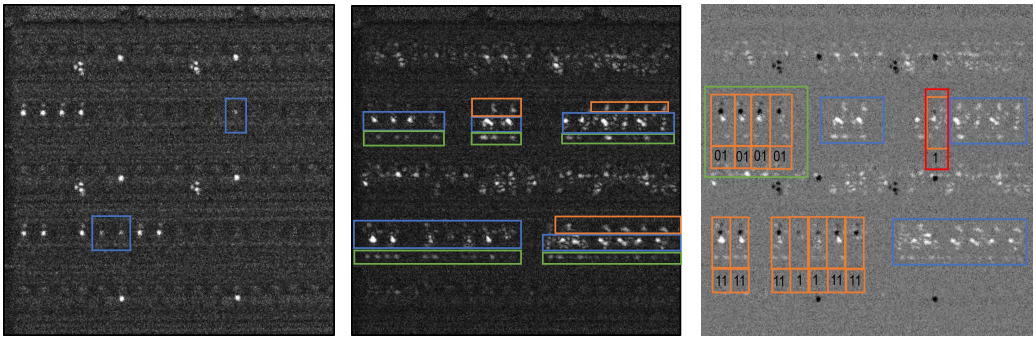
6.2 Key Extraction from PoC Circuit Implementation

6.2.1 Adversary without Access to the Layout

In this subsection, we present how an adversary without access to the GDS II 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. 8a and 9a, respectively. The resulting bright nodes evident in that figure reveals the clock tree distribution and sequential logic elements over the chip. It is notable that in the profiling stage, it has been confirmed, each block shown in Fig. 5c carries two flip-flops. The number of active flip-flops can be identified from the brightness shown in EOFM clock activity (Fig. 8a and Fig. 9a). 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. 8a and Fig. 9a.

Detecting the Key-registers: Once the EOFM for clock tree is identified, the



(a) (b) (c)

Figure 9: 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.

attacker can apply the approach described in Sect. 4.3 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. 8b and Fig. 9b are collected by rebooting the chip in a continuous loop. These measurements contain the activity of both sequential and combinatorial logic elements. In Fig. 8b and Fig. 9b, the blue, green and orange rectangles represents the logic elements (both sequential and combinatorial), input buffers and output buffers, respectively. To aid the mapping of two different frequencies, in PHEMOS, an attacker can subtract the EOFM measurement for resetting frequency from clock frequency as shown Fig. 8c and Fig. 9c. In both of the images, the black and white dots represent the elements active at the clock and reset frequency respectively. These images expose the location of sequential elements and registers to the attacker.

Extracting the Key: Applying the method described in Sect. 4.4, 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. 8c and Fig. 9c. 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. 9c shows activity in the free-running mode. Therefore the remaining one flip-flop is identified as the register responsible for continuous resetting of the circuit.

Automation in the Detection Process: An adversary without having the access to layout can also apply image processing techniques to reduce the time and effort required for detecting key-register locations. She can detect registers locations in an automated fashion by comparing the EOFM activity of $\mathbf{x_0}$ and $\mathbf{x_1}$. The EOFM images collected from PHEMOS 1000 are RBG images. Therefore, the images are converted into the grayscale image followed by image pre-processing. First, The images are denoised with a median filter. Gaussian low pass filter is applied to sharpening the edges of the images. The resulted images are binarized using thresholding. Fig. 10a and Fig. 10b shows the resultant pre-processed images for $\mathbf{x_0}$ and $\mathbf{x_1}$, respectively. Image pre-processing is followed by an intensity-based image registration with multimodal registration algorithm. Such registration method is widely used for X-ray, magnetic resonance images. As the image acquisition parameters vary from one EOFM to another EOFM image, variation is present

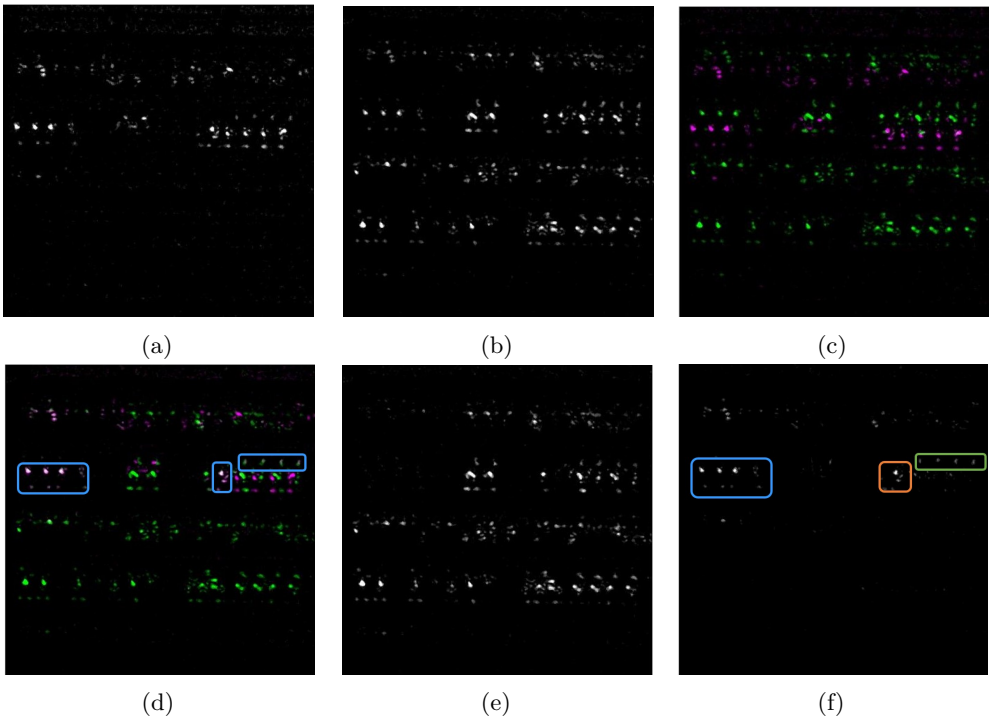
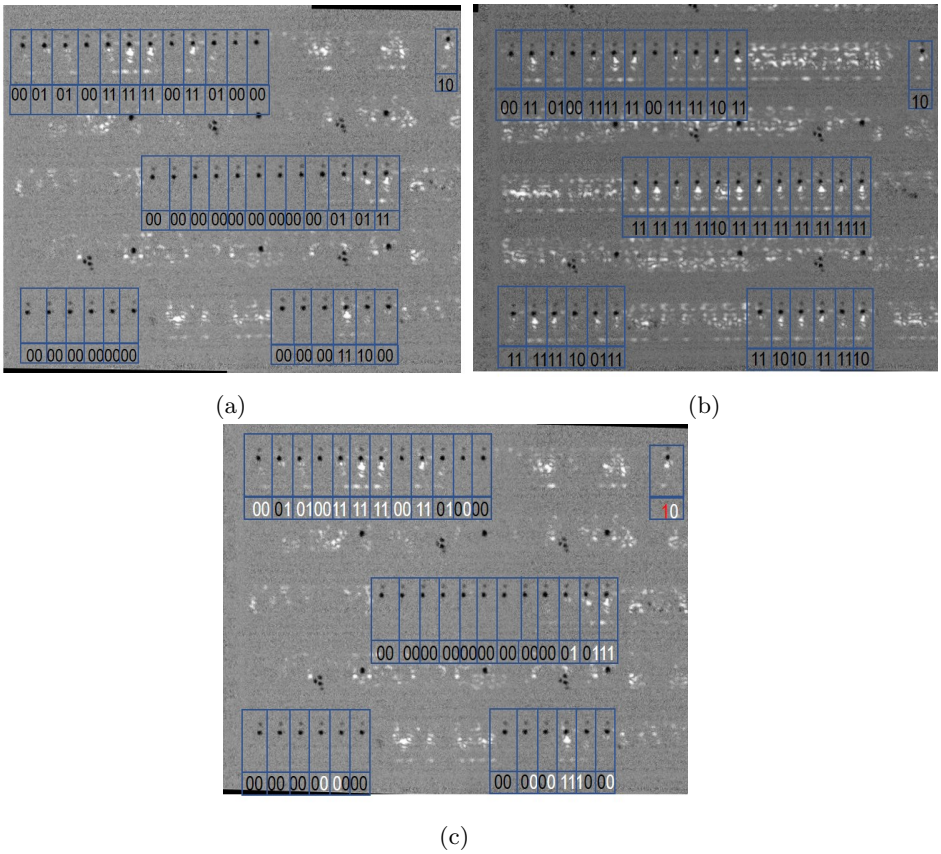


Figure 10: Comparing EOFM activity of $\mathbf{x_0}$ and $\mathbf{x_1}$ for Localizing key-registers/gates through image processing (a) Pre-processed EOFM image of Fig. 8b; (b) Pre-processed EOFM image of Fig. 9b; (c) Overlay image of Fig. 10a and Fig. 10b before image registration where pink dots stands for logic element activity for $\mathbf{x_0}$ and green dots for logic element activity for $\mathbf{x_1}$; (d) Overlay image of Fig. 10a and Fig. 10b after image registration where squared areas shows common activity nodes for both Fig. 10a and Fig. 10b; (e) Difference between the logic element activity of Fig. 10a and Fig. 10b; (f) Common active nodes for both Fig. 10a and Fig. 10b and possible locations for keys.

in the images modality. Therefore, multimodal image registration is used for registering the two images. Fig. 10c and Fig. 10d compares the results before and after registration, respectively. In both cases, the images with the pink and green activity spots stand for $\mathbf{x_1}$ and $\mathbf{x_0}$, respectively. The bright dots (blue rectangles) in Fig. 10d show the points those matches with each of the source images i.e., Fig. 10a and Fig. 10b. Next, image subtraction has been applied to EOFM images. The resultant image is shown in Fig. 10e detects the nodes triggering with change in input vector. Now comparing Fig. 10e with Fig. 10a we can identify the nodes whose activity do not depend on the input vector. Now, we can identify locations enclosed as blue and green rectangles as key-registers and input buffers for key-gates respectively. The flip-flop with orange rectangle in Fig. 10f is the register connected with reset signal of the circuit.

6.2.2 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 netlist. Once the key-registers are localized, extracting key value requires only measuring the EOFM/EOP activity of those registers. Therefore, she can apply any input vector to the chip and measure the EOFM activity in a reset loop as shown in Fig. 9. By analyzing the EOFM activity of key-register in Fig. 9b or Fig. 9c, she can extract the key value using the approach described in Sect. 4.4.



(a) (b)

(c)

Figure 11: 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. (a) Subtracted image of EOFM measurement for clock and reset frequency for $\mathbf{x_0}$. The stored data in each register for input $\mathbf{x_0}$ is mentioned at the bottom of the corresponding register; (b) Subtracted image of EOFM measurement for clock and reset frequency for $\mathbf{x_1}$. The stored data in each registers for input $\mathbf{x_1}$ is mentioned at the bottom of the corresponding register; (c) Detecting the key locations and extracting keys from EOFM activity showed in Fig. 11a and Fig. 11b. 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.

6.3 Key Extraction from Obfuscation Benchmark Circuit

The methodology described in Sect. 4 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 and 596 gates. The circuitry is locked by 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 activity of the complete circuit on the chip is covered in 2×2 matrix and stitched later. The clock frequency of the FPGA chip has already been determined as 50.14 MHz. 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. 11a and Fig. 11b represent the subtracted image of EOFM activity at clock frequency from reset frequency for above-mentioned input patterns. The white and black spots represent activity at reset frequency and clock frequencies, respectively. From the clock EOFM activity the location of the registers are identified and showed in blue rectangles in Fig. 11a and Fig. 11b. The

values stored in all the registers are probed (the stored data are presented in Fig. 11a and Fig. 11b). As discussed in Sect. 4, the register maintaining the same state irrespective to applied input patterns are identified as the key-registers. Hence, the state of the registers for the applied input pattern $\mathbf{x_0}$ and $\mathbf{x_1}$ are differentiated. We identified the registers storing values written in white color in Fig. 11c 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. 11c, is the reset signal which can be identified by running the chip in free-running mode and measuring EOFM at the clock frequency. An adversary with complete reverse engineering capability can smoothly localize the key-registers and directly probe those registers to uncover the secret keys.

7 Discussion

7.1 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 attacker.

Reset Counter in the Device: Optical probing requires several repeated measurements to acquire an adequate signal to noise ratio. 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 [LTK+18]. However, an attacker with access to gate-level netlist 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.

Determining Clock Frequency for EOFM: Determination of operating frequency is a vital step for detecting the sequential elements of the chip. An attacker can determine the chip frequency by focusing the EOP signal on any clock carrying buffer identified from the circuit layout and connecting the reflected light signal to spectrum analyzer incorporated into the PHEMOS system. She runs the spectrum analyzer with the conventional frequency sweep mode to find the exact chip frequency value. Moreover, the adversary can conduct power or EM analysis in the frequency domain to detect the internal oscillators' frequencies. Another crucial step for exposing key value is determining the bootup period. This can easily be calculated by measuring the time between starting chip to generating output for any specific input pattern. This key-register condition can also be compared with the inactive state of the chip when no inputs are fed to the chip.

Extracting the Exact Key Sequence: Determining the exact key sequence might be a challenge for an adversary without having access to the gate-level netlist. She can extract each key bits using the methodology described in Sect. 4. However, since she does not have any knowledge about the order of the registers, she might not be able to generate the key bits in the right order to unlock the circuit. In this case, only the key-register locations and the stored key bits in them are revealed, which still can be imposed as the threat of ransom for the IP owner.

7.2 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 [CB09]. The operation mode of the IP relies on the applied key value. Similar to combinatorial logic locking, FSM-based obfuscation also assumes that the keys are stored in a tamper-

proof/secure memory. Consequently, similar to our proposed attack, an adversary can break into locking scheme through reading out the key from the key-registers. Similarly, other logic locking schemes, such as Stripped-functionality logic locking (SFLL) [YSN+17] approach, which is the current state-of-the-art countermeasure against oracle-guided attacks, is vulnerable to our optical probing attack as well.

8 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 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 silicon active regions thermally, conventional photosensors fail to trigger during optical probing. On the other hand, the thermal stimulation introduces temperature and current variations in the circuit, which can influence circuits, such as ring-oscillators (ROs) [TFL+17]. In this case, the implementation of ROs as a probing protection scheme can be used to generate an anti-tamper reaction in the chip to protect the locking keys. In [SATF18] 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. As a solution, the IP owner can use 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.

9 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

[AK96] Ross Anderson and Markus Kuhn. Tamper resistance-a cautionary note. In *Proceedings of the second Usenix workshop on electronic commerce*, volume 2, pages 1–11, 1996.

[AK07] Yousra Alkabani and Farinaz Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX security symposium*, pages 291–306, 2007.

[ASX+18] Sarah Amir, Bicky Shakya, Xiaolin Xu, Yier Jin, Swarup Bhunia, Mark Tehranipoor, and Domenic Forte. Development and evaluation of hardware obfuscation benchmarks. *Journal of Hardware and Systems Security*, pages 1–20, 2018.

[ATF17] Navid Asadizanjani, Mark Tehranipoor, and Domenic Forte. Pcb reverse engineering using nondestructive x-ray tomography and advanced image processing. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 7(2):292–299, 2017.

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

[BGG+13] Lilian Bossuet, Michael Grand, Lubos Gaspar, Viktor Fischer, and Guy Gogniat. Architectures of flexible symmetric key crypto engines—a survey: From hardware coprocessor to multi-crypto-processor system on chip. *ACM Computing Surveys (CSUR)*, 45(4):41, 2013.

[BS13] Brent R Beachem and Merrill K Smith. Key management to protect encrypted data of an endpoint computing device, November 19 2013. US Patent 8,588,422.

[BTZ10] Alex Baumgarten, Akhilesh Tyagi, and Joseph Zambreno. Preventing ic piracy using reconfigurable logic barriers. *IEEE Design & Test of Computers*, 27(1), 2010.

[CB08] Rajat Subhra Chakraborty and Swarup Bhunia. Hardware protection and authentication through netlist level obfuscation. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 674–677. IEEE Press, 2008.

[CB09] Rajat Subhra Chakraborty and Swarup Bhunia. Harpoon: an obfuscation-based soc design methodology for hardware protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(10):1493–1502, 2009.

[Cor] Microsemi 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.

[CRT13] Gustavo K Contreras, Md Tauhidur Rahman, and Mohammad 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*, pages 196–203. IEEE, 2013.

[CSW16] Franck Courbon, Sergei Skorobogatov, and Christopher Woods. Reverse engineering flash eeprom memories using scanning electron microscopy. In *International Conference on Smart Card Research and Advanced Applications*, pages 57–72. Springer, 2016.

[GDT14] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. Counterfeit integrated circuits: detection, avoidance, and the challenges ahead. *Journal of Electronic Testing*, 30(1):9–23, 2014.

[HNT+13] Clemens Helfmeier, Dmitry Nedospasov, Christopher Tarnovsky, Jan Starbug Krissler, Christian Boit, and Jean-Pierre Seifert. Breaking and entering through the silicon. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 733–744. ACM, 2013.

[HP18] Max Hoffmann and Christof Paar. Stealthy opaque predicates in hardware-obfuscating constant expressions at negligible overhead. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(2):277–297, 2018.

[IEE] 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.

[JM07] Richard Wayne Jarvis and Michael G Mcintyre. Split manufacturing method for advanced semiconductor circuits, March 27 2007. US Patent 7,195,931.

[KKF+18] Benjamin Kollenda, Philipp Koppe, Marc Fyrbiak, Christian Kison, Christof Paar, and Thorsten Holz. An exploratory analysis of microcode as a building block for system defenses. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1649–1666. ACM, 2018.

[LTBS16] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. No place to hide: Contactless probing of secret data on fpgas. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 147–167. Springer, 2016.

[LTK+18] Heiko Lohrke, Shahin Tajik, Thilo Krachenfels, Christian Boit, and Jean-Pierre Seifert. Key extraction using thermal laser stimulation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 573–595, 2018.

[MG] Amrit Mundra and Hong Guan. Secure boot on embedded sitara™ processors. http://www.ti.com/lit/wp/spry305a/spry305a.pdf. Accessed: 2018-09-30.

[NSSO12] Dmitry Nedospasov, Jean-Pierre Seifert, Alexander Schlösser, and Susanna Orlic. Functional integrated circuit analysis. In *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*, pages 102–107. IEEE, 2012.

[N.V17] NXP Semiconductors N.V. Realizing Today's Security Requirements: Achieving End-To-End Security with a Crossover Processor. Technical report, NXP, 8 2017. Accessed: 2018-09-30.

[Pat01] Baiju V Patel. Method for securing communications in a pre-boot environment, December 4 2001. US Patent 6,327,660.

[Pho] Hamamatsu Photonics. Emission Microscopy: Phemos-1000. https://www.hamamatsu.com/resources/pdf/sys/SSMS0003E_PHEMOS1000.pdf. Accessed:2018-04-26.

[QCF+16] Shahed E Quadir, Junlin Chen, Domenic Forte, Navid Asadizanjani, Sina Shahbazmohamadi, Lei Wang, John Chandy, and Mark Tehranipoor. A survey on chip to system reverse engineering. *ACM journal on emerging technologies in computing systems (JETC)*, 13(1):6, 2016.

[RKM10] Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. Ending piracy of integrated circuits. *Computer*, 43(10):30–38, 2010.

[RPSK12] Jeyavijayan Rajendran, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of logic obfuscation. In *Proceedings of the 49th Annual Design Automation Conference*, pages 83–89. ACM, 2012.

[RSSK13] Jeyavijayan Rajendran, Michael Sam, Ozgur Sinanoglu, and Ramesh Karri. Security analysis of integrated circuit camouflaging. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 709–720. ACM, 2013.

[RST+18] M Tanjidur Rahman, Qihang Shi, Shahin Tajik, Haoting Shen, Damon L Woodard, Mark Tehranipoor, and Navid Asadizanjani. Physical inspection & attacks: New frontier in hardware security. In *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*, pages 93–102. IEEE, 2018.

[RZZ+15] Jeyavijayan Rajendran, Huan Zhang, Chi Zhang, Garrett S Rose, Youngok Pino, Ozgur Sinanoglu, and Ramesh Karri. Fault analysis-based logic encryption. *IEEE Transactions on computers*, 64(2):410–424, 2015.

[SATF18] Haoting Shen, Navid Asadizanjani, Mark Tehranipoor, and Domenic Forte. Nanopyramid: An optical scrambler against backside probing attacks. In *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*, page 280. ASM International, 2018.

[SLM+17] Kaveh Shamsi, Meng Li, Travis Meade, Zheng Zhao, David Z Pan, and Yier Jin. Appsat: Approximately deobfuscating integrated circuits. In *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*, pages 95–100. IEEE, 2017.

[SRM15] Pramod Subramanyan, Sayak Ray, and Sharad Malik. Evaluating the security of logic encryption algorithms. In *Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on*, pages 137–143. IEEE, 2015.

[STBF17] Bicky Shakya, Mark M Tehranipoor, Swarup Bhunia, and Domenic Forte. Introduction to hardware obfuscation: Motivation, methods and evaluation. In *Hardware Protection through Obfuscation*, pages 3–32. Springer, 2017.

[TFL+17] Shahin Tajik, Julian Fietkau, Heiko Lohrke, Jean-Pierre Seifert, and Christian 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*, pages 186–191. IEEE, 2017.

[TJ09] Randy Torrance and Dick James. The state-of-the-art in ic reverse engineering. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 363–381. Springer, 2009.

[TLSB17] Shahin Tajik, Heiko Lohrke, Jean-Pierre Seifert, and Christian 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*, pages 1661–1674. ACM, 2017.

[TNH+14] Shahin Tajik, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, and Christian Boit. Emission analysis of hardware implementations. In *Digital System Design (DSD), 2014 17th Euromicro Conference on*, pages 528–534. IEEE, 2014.

[XS18] Yang Xie and Ankur Srivastava. Anti-sat: Mitigating sat attack on logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[XSTF17] Xiaolin Xu, Bicky Shakya, Mark M Tehranipoor, and Domenic Forte. Novel bypass attack and bdd-based tradeoff analysis against all known logic locking attacks. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 189–210. Springer, 2017.

[YMSR17] Muhammad Yasin, Bodhisatwa Mazumdar, Ozgur Sinanoglu, and Jeyavijayan Rajendran. Security analysis of anti-sat. In *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*, pages 342–347. IEEE, 2017.

[YSN+17] Muhammad Yasin, Abhrajit Sengupta, Mohammed Thari Nabeel, Mohammed Ashraf, Jeyavijayan JV Rajendran, and Ozgur Sinanoglu. Provably-secure logic locking: From theory to practice. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1601–1618. ACM, 2017.