On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs

Shahin Tajik*,[1], Heiko Lohrke*,[2], Jean-Pierre Seifert[1], and Christian Boit[2]
[1]Security in Telecommunications, [2]Semiconductor Devices
Technische Universität Berlin, Germany
{stajik,jpseifert}@sec.t-labs.tu-berlin.de
lohrke@mailbox.tu-berlin.de, christian.boit@tu-berlin.de
* These authors contributed equally to this work

ABSTRACT

Modern Integrated Circuits (ICs) employ several classes of countermeasures to mitigate physical attacks. Recently, a powerful semi-invasive attack relying on optical contactless probing has been introduced, which can assist the attacker in circumventing the integrated countermeasures and probe the secret data on a chip. This attack can be mounted using IC debug tools from the backside of the chip. The first published attack based on this technique was conducted against a proof-of-concept hardware implementation on a Field Programmable Gate Array (FPGA). Therefore, the success of optical probing techniques against a real commercial device without any knowledge of the hardware implementation is still questionable. The aim of this work is to assess the threat of optical contactless probing in a real attack scenario. To this end, we conduct an optical probing attack against the bitstream encryption feature of a common FPGA. We demonstrate that the adversary is able to extract the plaintext data containing sensitive design information and intellectual property (IP). In contrast to previous optical attacks from the IC backside, our attack does not require any device preparation or silicon polishing, which makes it a non-invasive attack. Additionally, we debunk the myth that small technology sizes are unsusceptible to optical attacks, as we use an optical resolution of about 1 µm to successfully attack a 28 nm device. Based on our time measurements, an attacker needs less than 10 working days to conduct the optical analysis and reverse-engineer the security-related parts of the hardware. Finally, we propose and discuss potential countermeasures, which could make the attack more challenging.

KEYWORDS

Bitstream Encryption; FPGA Security; Electro-Optical Probing; Laser Voltage Probing

1 INTRODUCTION

Several countermeasures have been integrated into modern Integrated Circuits (ICs) to protect the secrets and Intellectual Property (IP) from physical attacks, such as side-channel analysis and fault attacks. Counterfeiting and overbuilding of target products are the primary motivation behind these attacks [23, 25]. Recently, a new class of physical attack, relying on a known Failure Analysis (FA) technique, has been introduced [7], which is capable of circumventing the protections to get access to the secrets and IPs on the chip. This attack, which is called optical contactless probing, enables an adversary to probe volatile and on-die-only secret data from the backside of a chip without making any physical contact with transistors. It has been demonstrated that with the help of this technique, an attacker can localize and probe secret keys on a Field Programmable Gate Array (FPGA), which are required to decode the encrypted configuration binary data, called *bitstream*. However, the effectiveness of this attack has been evaluated against a proof-of-concept FPGA implementation, where the details of target implementation were known to the attackers. Furthermore, the technology of the chip selected as the target was 60 nm, which is larger than the technology of the latest generations of ICs. This raises the question if this technique can still be applied in a real attack scenario, where little or no knowledge about the underlying hardware implementation is available to the attacker. Moreover, it is unclear, whether optical probing can be applied to recent IC technologies as well.

The primary aim of this work is therefore to assess the threat of optical contactless probing against a *real* modern commercial device, where an adversary has only access to the publicly available documentation. In other words, it would be interesting to evaluate the feasibility of extracting sensitive information by an attacker, who possesses the target platform and is capable of renting the necessary equipment from a failure analysis lab for a limited time to launch an attack. To this end, an appropriate target device with strong security features has to be chosen. FPGAs seem to be suitable targets as they have become indispensable parts of embedded electronic devices in several applications such as cryptography, digital signal processing and Software Defined Radios (SDRs). Moreover, they are deployed in the switches of Software Defined Networks (SDNs) and considered as the primary components of the Centralized Radio Access Network (C-RAN) concept in 5G cellular networks. Besides, internet giants have already integrated FPGAs into their cloud computing platforms to provide customers with more flexible and faster services [1, 10]. Therefore, a great deal of attention has to be paid to protect the secrets and IPs on these platforms from cloning.

Since SRAM-based FPGAs do not contain any internal Non-Volatile Memory (NVM) to store the bitstream, the bitstream is loaded in an untrusted field from an external NVM to the device upon each

power-on. Even flash-based FPGAs capable of storing the configuration inside their packages might be reconfigured by a remote server, which leads to the transmission of updated bitstreams in an adversarial environment. If an unauthorized person can obtain the bitstream, she might be able to create a counterfeit product by cloning and reverse-engineering the design. Bitstream encryption is a conventional solution used by several FPGA vendors to assure the confidentiality of the bitstream. To implement this feature, FPGAs contain state of the art decryption cores in the form of Application-Specific Integrated Circuits (ASIC) to protect the bitstream data.

We have chosen a device from Xilinx's 7-Series FPGAs manufactured with 28 nm technology to present our attack. Selection of this device allows for responsible disclosure and prevents additional harm to device security, as 7-Series bitstream encryption has already been broken by performing side-channel analysis [5, 15]. It should be noted that the divulged information from the published attack in [5, 15] only considers the innermost AES workings, and therefore, offers no helpful information for the attack that will be evaluated in this work. Needless to say that during our experiments, we did not possess any additional knowledge other than publicly available information about the device and the ASICs contained therein. Additionally, the technology of this FPGA is representative for several modern ICs, and it is small enough to reveal the strength of our approach.

Our Contribution. We present how an attacker can in a *non-invasive* manner and without *any device preparation* localize the bus, which is connected to the output of the decryption core and responsible for carrying and distributing the plaintext bitstream on the chip. We further demonstrate that after finding the bus, the attacker can probe the passing bitstream information on it directly and reconstruct the bitstream data offline. Based on the achieved results, it becomes apparent that if no proper protection is provided by vendors, the same attack can also be applied to the latest generation of FPGAs, which are *thought to be secure*. Additionally, we reveal the time that was needed to mount our attack successfully. The required time and effort are shown to be much less than what was expected, i.e., less than 10 working days use of failure analysis equipment. Finally, we discuss potential countermeasures, which can be implemented by vendors as well as users to protect the secrets and IPs on their chips against optical contactless probing.

2 BACKGROUND

2.1 Optical Contactless Probing

Optical techniques have been developed in the field of failure analysis to debug ICs in a contactless way. Contactless interaction with the transistors requires less effort in comparison to other debugging tools, such as Focused Ion Beam (FIB) circuit editing. While the optical path from the transistors to the surface of the IC is obstructed by multiple interconnected layers, the analysis can be carried out from the IC backside through the silicon substrate. However, silicon is only transparent to photons in the near infrared (NIR) spectrum.

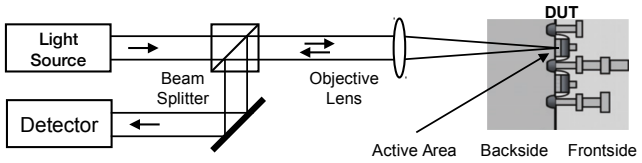


Figure 1: Simplified illustration of contactless optical probing signal acquisition.

Hence, the necessary equipment for the analysis of the IC, including the light sources and detectors, have to be designed for functioning in the NIR region.

Electro-Optical Probing (EOP) and Electro-Optical Frequency Mapping (EOFM) are examples of optical contactless techniques, which enable us to debug nanoscale transistors from the backside of the chip. Instead of the lasers used in Laser Voltage Probing (LVP) and Laser Voltage Imaging (LVI) techniques they use incoherent light sources. Apart from this difference, both techniques provide the same functions, and LVP/LVI and EOP/EOFM can be seen as equivalent tools. 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 silicon substrate from the IC backside to reach the transistors, which leads to a partial absorption and a partial reflection of the light. In the case of EOP, the reflected light is modulated based on the electrical signal on a node, and it can be fed to an optical detector to measure its intensity, see Fig. 1. In this way, the data passing through a node can be probed. Since the light modulation is small, the signal needs to be measured several times and averaged by running the device in a triggered loop. In this case, a sufficient signal to noise ratio can be achieved.

For performing EOFM, on the other hand, the detector signal is fed into a spectrum analyzer acting as a narrow band frequency filter while the light beam scans the device. In this case, the signal is not averaged. The beam is scanned across the Device Under Test (DUT) using galvanometric x/y mirrors, and the filter output of the spectrum analyzer is sampled for every scanned pixel. Subsequently, a control PC is used to assemble the sampled frequency filter values into a 2D image using a grayscale or false color representation. If an electrical node operates at the frequency of interest, it will modulate the light reflected with the same frequency, which will be able to pass through the frequency filtering spectrum analyzer. As a result, the nodes with a switching frequency equal to the frequency filter show up as bright spots in the EOFM image leading to their localization on the chip.

2.2 FPGA Security during Configuration

FPGAs are programmed and configured by binary data called the *bitstream*, which is generated by an application designer. While flash-based FPGAs have internal Non-Volatile Memory (NVM) to store configuration data in the same package, SRAM-based FPGAs do not contain any NVM, and hence, are not capable of storing the bitstream [24]. Therefore, the bitstream has to be kept in an external

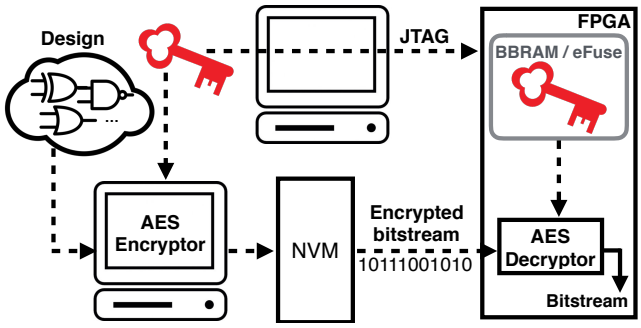


Figure 2: Encrypting the bitstream in the IDE using the key k and decrypting it on the FPGA by an ASIC decryption core using the same key.

NVM and loaded into the SRAM-based FPGAs upon each power-on in an untrusted field. Similarly, if the firmware of flash-based FPGAs requires configuration updates, their upgraded bitstream has to be transferred remotely to the device in a potentially adversarial environment. Transmitting bitstreams in plaintext can divulge the designs and IPs to an adversary. Consequently, bitstreams have to be kept confidential. Bitstream encryption is a common feature of modern FPGAs to prevent IP piracy during FPGA configuration.

2.2.1 Bitstream Encryption. To enable bitstream encryption, a secret key k is used to encrypt the application design in the Integrated Development Environment (IDE) software. While recent generations of FPGAs from Xilinx, Intel/Altera, and Microsemi deploy AES-256 to encrypt the bitstream, the mode of operation might differ on them. Xilinx 7-Series FPGAs are using AES in Cipher Block Chaining (CBC) mode to encrypt the bitstream [27]. In this case, the bitstream is divided into n 128-bit blocks $p^{i \in \{1,\cdots,n\}}$ and the resulting encrypted bitstream in n 128-bit blocks is generated by

$$c^i = AES_k^{ENC}(p^i \oplus c^{i-1}),$$

with $c^0 = IV$ (i.e., initialization vector). The key k is transferred to the FPGA via JTAG in a safe environment and the encrypted bitstream is stored in an external NVM, see Fig. 2. The transferred key on the FPGA is stored either in the Battery Backed RAM (BBRAM) or eFuses inside the FPGA. Each time the FPGA is powered up in the untrusted field, the encrypted bitstream is transmitted to the chip, and it is decoded by a decryption core using the stored key k inside the chip. In this case, the plaintext (i.e., unencrypted bitstream) is generated by

$$p^i = AES_k^{DEC}(c^i) \oplus c^{i-1},$$

with $c^0 = IV$.

2.2.2 Bitstream Authentication. Xilinx's 7-Series FPGAs employ authenticated encryption schemes to assure confidentiality, integrity and authenticity of the bitstream [27]. To authenticate the bitstream, Hash Message Authentication Code (HMAC) is used. In this scheme, HMAC is performed with an authentication key K_a (not to be confused with the encryption key k) on the unencrypted bitstream. However, in contrast to the encryption key, there is no storage on the FPGA for the HMAC key. Therefore, the key and the MAC itself is encrypted with the bitstream using the encryption key k and sent to the FPGA. As a result, the correctness of the bitstream can be checked only when the whole bitstream, including the K_a and the MAC, is already decrypted.

2.2.3 Attacks against FPGAs. Here we briefly review the published successful attacks against dedicated ASIC decryption cores of FPGAs. Side-channel analysis is the primary technique deployed against dedicated ASIC security circuits of FPGAs in the literature. In one of the first attempts, side-channel analysis was used to discover a keyed backdoor/test mechanism in a Microsemi FPGA, which activated the readback of the bitstream in plaintext after configuration [20]. In subsequent attempts, side-channel analysis was mainly employed to attack the bitstream encryption feature of FPGAs to extract the secret key k. It has been demonstrated that the security of DES and AES decryption cores of the Xilinx and Intel/Altera FPGAs can be broken by performing differential power analysis (DPA) [12–14, 21]. However, although DPA is a non-invasive attack, it requires custom boards to reduce the noise of the measurement, which makes the attack more challenging. To mount an attack against the bitstream encryption without any board modifications, electromagnetic analysis (EM) can be deployed [5, 15]. Side-channel analysis can be mitigated by using asymmetric authentication, key rolling and side-channel resistant decryptors. These security schemes have been already integrated into the most recent FPGA generations [9, 17].

3 APPROACH

This section describes the approach employed in this work to assess the threat of contactless optical probing towards commercial security ASICs. To be more specific, we chose a Xilinx Kintex 7 FPGA for our experiments. For the general attack scenario we assume the following: The attacker has physical possession of a board containing an FPGA in a modern flip-chip package which loads an encrypted bitstream from NVM. She seeks to extract the plaintext bitstream data to reveal the IPs or secrets (e.g., authentication keys) contained therein. However, the only professional equipment she has access to is an optical probing system, which she rents at a failure analysis lab. Apart from that, she only has access to conventional equipment like a laptop and a soldering iron.

To evaluate an actual attack, first we have to consider how an attacker would proceed. In general, to enable plaintext extraction, she would need to perform the following basic steps:

(1) Localize the general configuration logic area on the silicon die

(2) Localize the AES decryption core in the configuration logic

(3) Localize the logic gates carrying plaintext data in the AES

(4) Extract the data from the found plaintext gates

To be on the safe side, the attacker would probably perform all these steps on a "training" device identical to the one she tries to attack.

This would allow her to feed the training device with manipulated bitstreams, as she can erase and manipulate the keys and security settings in this device freely. As soon as she finds the plaintext gates and is confident that her attack is working, she can move on to extract the data from the actual target.

Most of these steps are rather straightforward if she has access to an optical probing system. As the device is in a flip-chip package which exposes the silicon backside directly, she can use the common 1.3 µm optical probing wavelength to acquire reflected light images. As silicon is transparent to this wavelength, this will deliver images of the circuit structures *without any chip preparation*. Just from the first reflected light images of the device, she should already be able to distinguish between FPGA user logic fabric and ASIC areas. As the FPGA logic fabric consists of several rows and columns of identical elements, such as Custom Logic Blocks (CLBs) and memory cells, it has a highly ordered appearance. ASIC areas, on the other hand, are composed of different blocks for individual sub-functions and synthesized logic areas, and therefore, posses a more irregular structure. As soon as the attacker has identified ASIC candidates in this way, she can start to analyze them globally for identical appearance. If there are identical ASIC areas, they are unlikely candidates for the configuration ASIC, as it is expected that there is only one ASIC for this function. Apart from that, she might also find helpful information by comparing the functions mentioned in the datasheet to the structure and placement of the ASIC areas in question.

The remaining candidates can then be examined using optical probing. If the attacker can estimate some specific frequency present in the configuration logic, she might perform EOFM to detect the logic gates operating at this frequency. If this is not possible, she might deliberately induce a frequency by manipulating the bitstream data and detect it via EOFM. When she identifies the configuration logic, she can then compare the activity in this area for encrypted and unencrypted bitstreams. Areas that are only active for encrypted bitstreams are potential candidates for the decryption logic. In these areas, she will then need to somehow distinguish the logic carrying the plaintext from all other gates.

This distinction will require a slightly more sophisticated approach. Many modes of operation for block ciphers, especially AES-CBC in our case, destroy structures and frequencies present in the plaintext when they create the ciphertext. This is a valuable property of a cipher as it renders a number of attacks (e.g., frequency analysis) ineffective. Through this process, the spectrum of the ciphertext basically becomes computationally indistinguishable from noise in the frequency domain, see Fig. 3. However, during decryption, the plaintext structure is obviously fully recreated. Therefore, a frequency induced into the plaintext bitstream data would vanish in the ciphertext, and only reappear in the plaintext that leaves the decryption core. For instance, to induce a desired frequency for a signal leaving the AES decryption core, one can generate a time-periodic plaintext with a regular "10" pattern. Thus, we can write the plaintext $p(t)$ as a square wave:

$$p(t) = 2[H(t/T) - H(t/T - 1)] - 1$$

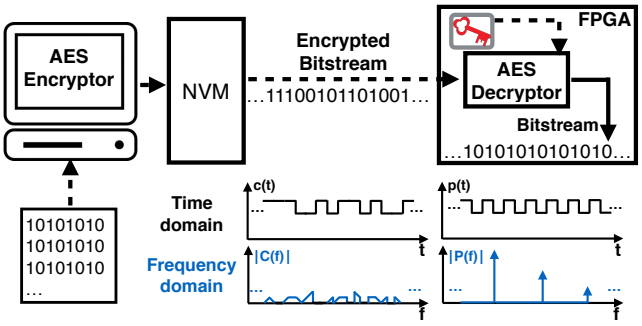


Figure 3: Since the ciphertext function $c(t)$ is indistinguishable from a random noise signal, its spectrum $C(f)$ contains no dominant frequency components. However, the plaintext $p(t)$ is generated as a periodic function, and hence, its spectrum $P(f)$ contains specific harmonics.

where $H(t)$ is the Heaviside step function [26] and T is the bit duration time. By expanding the periodic function $p(t)$ in terms of sum of sines [26], we have

$$p(t) = \frac{4}{\pi} \sum_{n=1,3,5,\cdots}^{\infty} \frac{1}{n} \sin\left(\frac{n\pi t}{T}\right).$$

This implies that only certain harmonics (i.e., frequency of sine functions) are included in the spectrum of the plaintext, see Fig. 3. In other words, if the attacker sets the EOFM filter to the fundamental frequency of $p(t)$, the logic carrying the decrypted plaintext data can be found while all nodes carrying the ciphertext data will generate no signal. Therefore, if the attacker seeks to identify the plaintext carrying logic, she simply needs to transfer an encrypted bitstream, manipulated to contain a certain frequency in its plaintext, onto the device. If she carefully chooses a frequency, which is unlikely to be present in other parts of the circuit, all gates carrying the plaintext can then directly be identified by performing EOFM at this frequency. Extracting the data from the logic gates found thus can then be attempted using EOP which constitutes the last part of the attack.

For all manipulated bitstreams that the attacker generates, she has to keep in mind that certain aspects of the decryption core implementation might alter the induced frequency. Nevertheless, she can solve this systematically, by creating a model of the device and deducing the generated frequencies from it. As an example, let us consider bitstream data containing alternating ones and zeros, which is loaded into a serial input using a configuration clock (CCLK). All logic gates carrying the serial data will then generate an EOFM signal at the fundamental frequency of CCLK/2. However, if we assume that at some point the same data is loaded onto a 32-bit parallel bus, the ones and zeroes are aligned on the individual bus lines and the signal on each line is static. In this case, no fundamental frequency is generated at all, and therefore, the logic gates of the data bus cannot be detected. However, if the attacker takes this parallel bus into account, she can generate bitstream data that contains 32 ones followed by 32 zeros. This leads to all bus lines being toggled for every 32-bit word transferred on the

Figure 4: Image of a Kintex 7 XC7K70T device in a flip-chip BGA package [6]. The exposed silicon backside of the die can be seen in the middle of the package.

bus. If the input into the device still functions in a serial fashion with the CCLK clock, the fundamental frequency of the parallel bus lines is CCLK/2/32 or CCLK/64 and they can be detected using EOFM. Using a model for prediction of frequencies and testing it using EOFM allows the attacker to determine if her current model is correct. Hence, she can gradually develop a model that considers all implementation details relevant to EOFM into account and generate a matching bitstream.

As a probable attack path is outlined, the only remaining aspect of the approach is the estimation of the effort for developing and executing the attack. As we expect the optical probing system to be the most costly factor of the attack, we will use time tracking software on it while attempting the attack discussed previously. This will allow us to give a precise measure of the time needed on the machine, for which in our scenario the attacker has to pay for. Nonetheless, we do not try to assess the amount of time spent for planning, programming and soldering, as we assume that this is not a limiting factor for the attacker. If the attack is successful, we can afterwards state the precise number of hours on the optical probing setup that have been needed to develop the attack. Furthermore, we can estimate how long it would take the attacker to execute plaintext data extraction when she already has knowledge of all relevant locations.

4 EXPERIMENTAL SETUP

This section describes the setup used to realize the approach discussed in Sect. 3.

4.1 Device Under Test

We chose a Skoll Kintex 7 FPGA development board designed by Numato Lab as the target platform. It contains a Xilinx XC7K70T Kintex 7 FPGA manufactured with 28 nm technology in a flip-chip Ball Grid Array (BGA) package, see Fig. 4. 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 therefore, we have direct access to the silicon substrate on the backside of the chip. Based on our measurements, the thickness of the substrate is about 700 µm. By

(a) (b)

Figure 5: (a) Reflected light overview image of the complete XC7K70T FPGA die. (b) Zoomed-in view of (a), showing repeating FPGA logic fabric structures (left half) versus more irregular dedicated function blocks (right half).

selecting a light source with a wavelength to which the silicon is transparent (1.3 µm), an image of the die can be acquired without any substrate thinning, see Fig. 5a. Hence, to conduct an optical attack from the backside of the chip, no preparation is required.

4.2 Electrical Setup

The FPGA on the development board can be configured either directly via JTAG or by loading the bitstream data from an on-board flash. The flash memory is connected via an SPI bus and has 128 Mbit capacity. Programming of the flash memory is performed through USB which is handled by an FTDI FT2232H chip. The USB connection also supplies the board power. As the FPGA configuration time is reduced significantly when using flash memory, we have chosen this scheme for our experiments. The board is designed to use the "Master SPI Configuration Mode", see [28]. If this mode is used with standard settings, the FPGA requests the data via a 1-bit wide SPI bus from the flash memory during configuration. To do so, it generates and outputs a clock signal on its CCLK pin which is used as the SPI bus master clock (Also called SCLK). It then issues a read instruction via the SPI bus and receives the bitstream data through its data input (DIN) pin which is connected to the "master in slave out" (MISO) pin of the flash [30]. The FPGA checks the received data for validity, and if no errors are found during configuration phase, it switches to user mode.

The only modifications to the development board are the following: We soldered coaxial cables to the CCLK output and DIN of the FPGA. This allowed us to monitor the data entering the FPGA as well as provide robust access to the bitstream data clock. We added an additional cable to the PROGRAM_B pin of the FPGA,

which if pulsed low triggers device reconfiguration. PROGRAM_B can be controlled manually via a switch or automatically through a function generator (Rigol DG4162). Additionally, we disabled the on-board switched-mode power supply for the 1.0 V net and replaced it with an external power supply (Agilent E3645A) as it was generating increased noise.

This setup can then be used to trigger configuration repeatedly using the function generator. In combination with a manipulated bitstream, it allows for continuous fundamental frequency generation (see Sect. 3) for EOFM. For triggering EOP waveform acquisition, either the CCLK, PROGRAM_B, or even the DIN signal can be used.

4.3 Optical Probing 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). Moreover, the setup uses an Advantest U3851 spectrum analyzer for EOFM while EOP waveforms are acquired using a LeCroy WavePro 735Zi oscilloscope. Three objective lenses were used during this work: 5x/0.14 NA, 20x/0.4 NA, 50x/0.71 NA. The 50x lens is equipped with a correction ring for silicon substrate thickness. The light source supplies the following maximum amounts of power onto the Device Under Test (DUT) with each objective lens: 5x: 63 mW; 20x: 26 mW; 50x: 45 mW. 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 oscilloscope 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.

4.4 Fake Bitstream Generation

To automate the generation of manipulated bitstreams as discussed in Sect. 3, we implemented a Python script with about 400 lines of code. This script uses the "pycrypto" package [4] to provide encryption and decryption with AES-CBC. The basic structure and fields of a regular encrypted Kintex 7 bitstream can be seen in Fig. 6a. In this section, we will only explain the fields relevant to our experiments and refer the reader to [24, 28] for more details. Using the Python script, the ciphertext portion of a regular bitstream can be extracted and decrypted, see Fig. 6a. In this decrypted portion, the "FDRI" block contains the actual configuration data, which makes up most of the bitstream size. This block is then replaced with a user-defined pattern, see Fig. 6b. Additionally the "footer commands" block is overwritten with "no operation" (NOP, 0x20000000) commands. As the footer commands would trigger FPGA startup, this prevents the device from enabling operation with the fake data after the configuration is done, avoiding potential device damage. Afterwards, this data is re-encrypted and the fake ciphertext placed back into

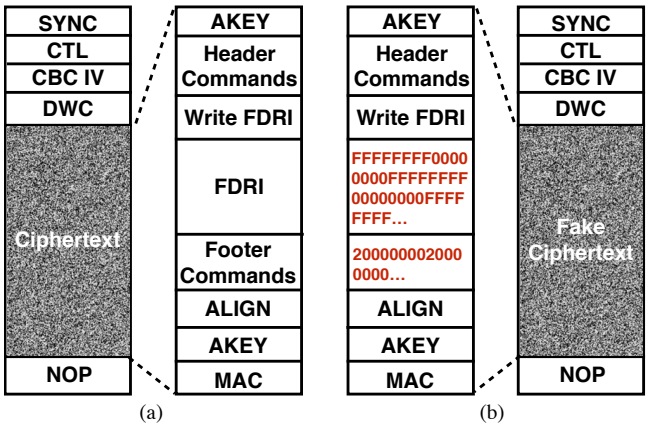


Figure 6: Xilinx Kintex 7 bitstream structure. (a) Regular bitstream [24, 28]. (b) Manipulated bitstream for EOFM fundamental frequency generation.

the regular bitstream structure, see Fig. 6b. This data can then be loaded into the on-board flash of the board and will be used by the FPGA for configuration. For all of our experiments we set the same AES key. Besides, the script performs some helper functions like looking up bitstream commands documented in the datasheet [28] and generating a human readable bitstream analysis. It can also preview the wraparound of the configuration data pattern under the assumption of different bus widths.

5 RESULTS

This section presents the results achieved applying the approach presented in Sect. 3 with the setup of Sect. 4. For visualizing EOFM activity maps, we have chosen to overlay the EOFM data onto reflected light images to aid orientation, (e.g., Fig. 8). The EOFM data is encoded in green, while the reflected light image is grayscale encoded, although with reduced brightness. This is to allow readers with a black and white representation to still distinguish EOFM and reflected light data by intensity. The threshold of EOFM data has been set slightly above the noise level to remove background noise and only show locations generating a signal at the set EOFM filter frequency.

5.1 Localization of the Configuration Logic

Fig. 5a shows a reflected light overview image of the whole die acquired with 1.3 µm wavelength. The image has been composed of multiple 5x objective measurements using image stitching software [18]. As the chip is not thinned, the die markings are also visible. The text is mirrored in this case, as the PHEMOS software automatically flips the image data in through-silicon observation mode. The image shows differences in the general layout of the appearing structures. On the one hand, there are very regular structures which consist of identical elements, see Fig. 5b left. These appear as regular vertical bars in the overview image of Fig. 5a and

Figure 7: Reflected light image of the configuration logic area. The image is 90 degrees tilted with regard to Fig. 5a.

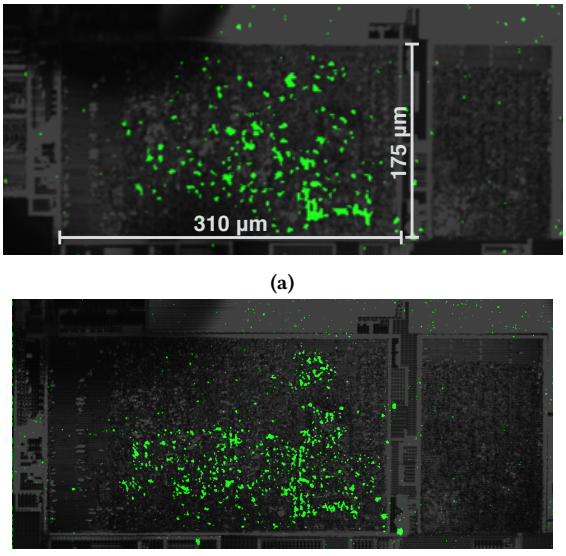


Figure 8: Comparison of CCLK activity for different bitstream settings in the region that always showed activity. (a) Unencrypted bitstream. (b) Encrypted bitstream. Only minor changes in activity can be observed. As this region is always active, it is a candidate for the main configuration logic.

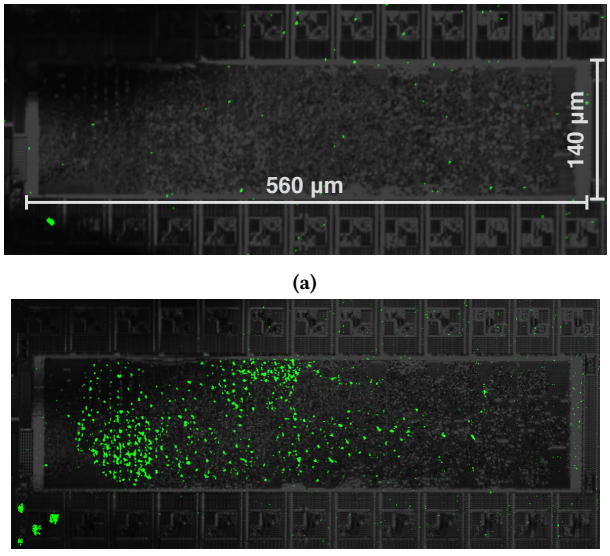


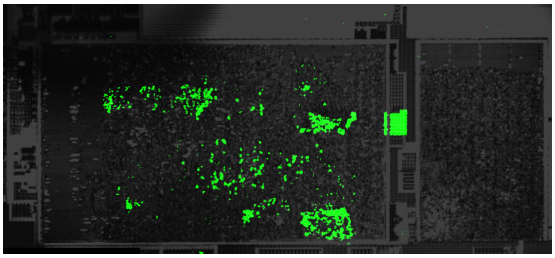
Figure 9: Comparison of CCLK activity for different bitstream settings in the region that only showed activity when encryption was enabled. (a) Unencrypted bitstream. (b) Encrypted bitstream. Because of this behavior this region is assumed to be the AES decryption core.

can be seen in most of the die image. As the FPGA logic fabric consists of many identical configurable logic and memory blocks (CLBs and RAM), we can assume that the highly ordered areas contain these elements, and thus, do not contain the configuration logic. On the other hand, there are blocks which contain more irregular patterns, see Fig. 5b right. For instance, in the overview image of Fig. 5a these can be seen in the center strip and the upper right hand corner of the die. Comparison with Xilinx data sheets [29] shows the upper right hand corner blocks to be the GTH/GTX transceivers available in some Xilinx's 7-Series FPGAs. The same data sheet also shows the configuration logic as a block placed roughly in the middle of the FPGA die. Therefore, this area seems to be the best candidate for the location of the configuration logic. Fig. 7 shows a detailed image of this area, which has been acquired with the 20x lens and stitching. To confirm this assumption, EOFM at the configuration clock (CCLK) frequency was performed, to reveal logic operating on the serial input bitstream data. This measurement indeed revealed activity in the area of Fig. 7, and consequently, confirmed that it contains the configuration logic. Further experiments with different frequencies introduced into an unencrypted bitstream also allowed us to identify and probe the input logic for the serial bitstream data.

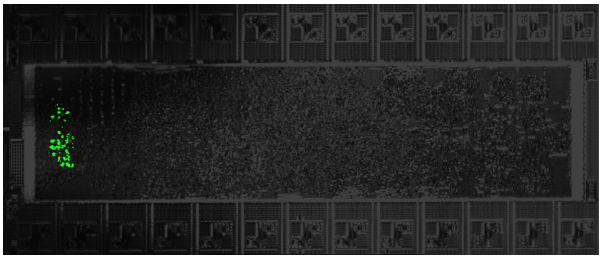
5.2 Localization of the AES Core

After the general configuration logic area was identified, the AES decryption core needed to be localized inside it. A standard IC design technique is to disable logic parts that are not used currently, either by clock gating or by completely powering them down. A comparison of activity with encrypted and unencrypted bitstreams should help to identify logic parts that are active only for encrypted bitstreams. These logic areas would then be strong candidates for the AES decryption core. To perform this comparison, EOFM measurements were acquired at the CCLK frequency in all of the configuration logic area shown in Fig. 7. For these measurements, the CCLK frequency was set to the "3 MHz" standard value. This indeed revealed an area which was always active, as well as an area that became active only for encrypted bitstreams.

A comparison for encrypted and unencrypted bitstreams is shown in Fig. 8 for the always active region and in Fig. 9 for the region only active for enabled encryption. As the region of Fig. 8 is always active, it probably contains the basic configuration logic and is



(a)



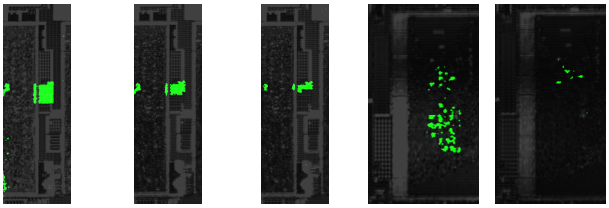
(b)

Figure 10: Activity map at the 32-bit word frequency (CCLK/64) revealing logic gates potentially connected to the 32-bit data bus. (a) "Main" logic area. (b) "AES" logic area. Although an unencrypted bitstream is used, the AES input logic is visible in (b).

therefore named the "main core". The region of Fig. 9 on the other hand is assumed to be the AES core.

5.3 Determination of the Bus Width

To identify the logic gates carrying actual data, as opposed to logic simply connected to the clock, the bitstream data was filled with alternating ones and zeroes. This would lead to all logic gates carrying the serial bitstream data to have a fundamental frequency of half the configuration clock (CCLK/2). However, EOFM measurements at this frequency showed only very minor activity. Hence, it was suspected that the data is parallelized and loaded onto a bus at some early stage in the configuration logic. The parallelization would change the fundamental frequency of the data, and would also require different data inserted into the manipulated bitstream for successful frequency generation, see Sect. 3. As the datasheet states that the bitstream data uses 32-bit words as its basic format [28], this was the first bus width that was considered. An unencrypted bitstream was prepared that contained a repeating pattern of 32 "1" bits and 32 "0" bits. This bitstream should then cause every bus line to flip its logic state for every word. The fundamental frequency of the bus lines would then be CCLK/2/32 or CCLK/64, which could then be detected by EOFM. Since the preamplifiers lower frequency limit is 100 kHz, we increased CCLK to the "12 MHz" setting. Note that this is not a limitation for the attacker, as at this stage she is working on the "training" device, and therefore, has full control of the bitstream, see Sect. 3. Additionally, to avoid damaging the FPGA by starting it with this manipulated bitstream, the startup



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

Figure 11: Comparison of activity at the 32-bit word frequency (CCLK/64) for a different number of active bits on the data bus. Suspected "main" logic output bus: (a) 32 bit active, (b) 16 bit active, (c) 8 bit active. Suspected "AES" logic input bus: (d) 32 bit active, (e) 8 bit active. This measurement has been acquired with decryption disabled to show only the input logic in the AES core.

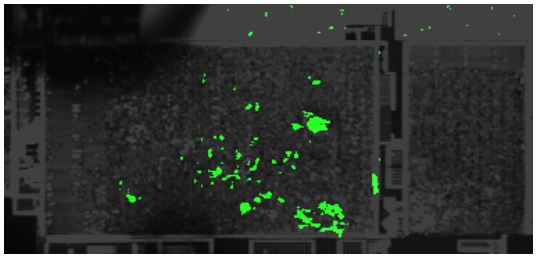
commands at the end of the bitstream data were replaced with "no operation" (NOP) commands.

The resulting EOFM measurement for the main core can be seen in Fig. 10a. It is evident that there are a many logic gates active in the main core at this frequency. This suggests that the 32-bit bus assumption is correct. Further test with different bus widths considered also supported the 32-bit bus hypothesis. It should also be noted that at the right hand side of Fig. 10a there is a rectangular area of activity directly at the edge of the active structure. Because of its placement and ordered appearance, this activity is a potential candidate for an input/output port. Curiously, even though encryption was disabled for this measurement, there was activity at this frequency in the AES core, see Fig. 10b. It is assumed that this activity is caused by the data input gates in the AES. The reasoning behind this is that if the data bus is directly connected to the AES, the data signal is always present in the very first stages of the input logic. Even if encryption is disabled, the data signal path would always be visible up to the first gate in the AES that requires a clock or enable signal for operation.

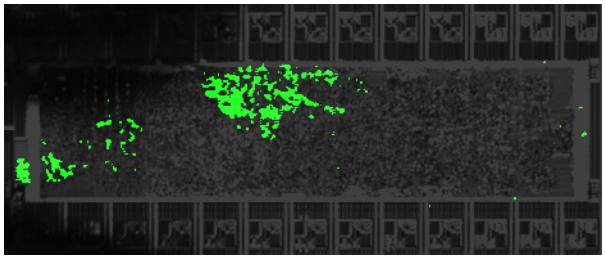
To further confirm the 32-bit bus hypothesis, the measurements were repeated with only some of the bus lines being active. For this, a portion of the bits in the 32-bit words was simply set to "always zero" while the others still were active with CCLK/64 as their fundamental frequency. The results of these measurements can be seen in Fig. 11, which all indicate a 32-bit bus. From this we concluded that the basic data bus width in the main core is 32-bit. It is also plausible that this bus width is the same for the AES input.

5.4 Localization of the Plaintext Output

To finally localize the logic gates carrying the decrypted plaintext data, the approach discussed in Sect. 3 was used. To this end, a bitstream with a data pattern containing the desired fundamental frequencies was generated, then encrypted, and finally transferred to the on-board flash memory. As these frequencies are absent in the ciphertext because of the encryption (see Sect. 3), they are only



(a)



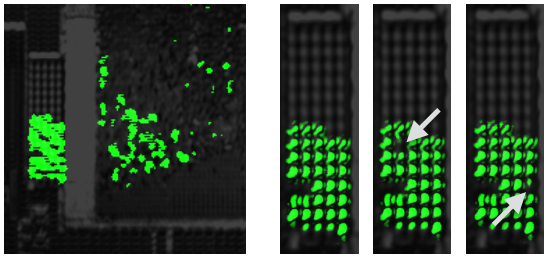
(b)

Figure 12: Activity map at the plaintext data frequency (CCLK/256) revealing gates potentially carrying the decrypted bitstream data. (a) "Main" logic area. (b) "AES" logic area. The leftmost edge of the AES area shows activity that might indicate an output port.

present in the plaintext after the AES decryption core. If the considerations regarding the influence of the bus width from Sect. 3 are observed, the plaintext gates should be visible in an EOFM measurement at the corresponding frequency. A first measurement assuming a 32-bit wide AES output demonstrated no activity. As AES is a cipher that operates on 128-bit blocks, another likely candidate for the AES output width was 128 bits. An EOFM measurement was performed using a repeating 128 "1" bits and 128 "0" bits pattern in the plaintext with an EOFM frequency of CCLK/2/128 or CCLK/256. As previously, because of the preamplifier, we increased CCLK, this time to the "33 MHz" setting.

The results of this measurement are shown in Fig. 12. There is a lot of activity in both the main core and the AES core which suggested that the 128-bit bus width assumption was correct. The activity at the plaintext data frequency in the main core also suggests that the data is fed back into the main core after decryption. This makes sense, as the Xilinx data sheets [28] indicate that the decrypted bitstream data is actually almost identical to an unencrypted bitstream. Hence, it would be efficient to use the same command and configuration data logic for all bitstreams, and simply route the input data through the AES core in case of an encrypted bitstream.

To allow actual data extraction, the next step was the selection of a suitable AES output port candidate and the determination of probing locations for every bit line of the bus. At the leftmost edge of Fig. 12b there is an area that bears similarity to the main core output port already shown in Fig. 11a. This area is also placed at the edge of the active structure and also has an ordered appearance, which

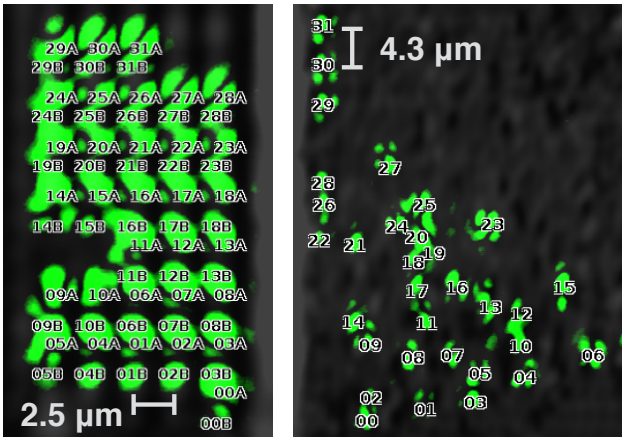


(a) (b) (c) (d)

Figure 13: Detailed activity map of the AES plaintext output area at the plaintext data frequency (CCLK/256). (a) Zoomed in measurement showing ordered "output port" logic (left) as well as unordered gates inside the logic mesh (right). Further zoomed in view of only the "port" area: (b) 32 bits active. (c) Bit 25 inactive. (d) Bit 13 inactive.

is especially apparent in a zoomed-in measurement, see Fig. 13a left. Therefore, it was chosen as the best candidate to determine the location mapping of the bit lines of the suspected 128-bit bus. To achieve this, fundamental frequency generation on the respective bus lines was activated or deactivated by modifying the plaintext bitstream data as in the previous experiments. Surprisingly, these measurements gave strange results, as most of the 128 bus lines considered during mapping did not generate EOFM signals. As it turned out, the AES output port area only showed activity when the bits of the 3rd 32-bit word of a 128-bit block in the bitstream configuration data were active. However, if single bits were active or inactive in this "special" word, they would directly enable or disable signal spots in the AES output port. An example of this behavior is given in Fig. 13b, 13c, and 13d. This suggests that there is a 32-bit bus, however for some reason it only shows activity for words 128 bits apart. Yet, a 32-bit bus is contradictory to our first measurement, which did not show activity at the 32-bit word frequency. Hence, our results seem to be inconsistent, as they indicate spatially that there is a 32-bit bus, while they indicate temporally that there is a periodicity that matches a 128-bit bus. This can not be explained by a straightforward model which assumes that the AES outputs the decrypted data at regular intervals, spread evenly across the time needed to process the next ciphertext block.

However, if we assume that the configuration logic following the AES immediately processes the data as fast as possible, the previous results can be explained. As can be seen from the device specifications [28], this assumption seems plausible, as the configuration logic should be able to process data at much higher speeds than in our experiment. This is because our board uses a simple serial data input, as opposed to the also available faster parallel data input schemes. Therefore, we can deduce that in this case, the logic following the AES actually has to wait for the next AES block to become ready, because this is limited by the serial input data rate. As soon as a new 128-bit plaintext block is ready, the configuration logic could then process four 32-bit command / configuration data words as fast as possible. Assuming a 32-bit output bus, this would mean that on every bus line there would be three bits in fast succession, while the fourth bit will stay on the output registers while the



(a) (b)

Figure 14: Complete mapping of plaintext bus bit locations for plaintext data extraction. (a) Locations in AES output port. (b) Alternative locations inside AES logic mesh.

AES is busy processing the next block. As soon as the next block is ready, this process would repeat, again with the last bit staying on the bus lines while the AES is busy. Taking only the last bits into account, this would lead to a 128-bit periodicity on the bus, or CCLK/2/128 if the bits are flipped for every 128-bit block. This also explains why the three "fast bits" would not appear in the EOFM measurements at CCLK/256. As they only stay on the registers for a short time, they hardly influence the fundamental frequency component at CCLK/256 at all. As a result, this model would be capable of explaining the behavior that was apparent in the previous measurements. To verify these assumptions we performed EOP to be able to analyze the waveforms actually present at the active spots seen in Fig. 13. Using a bitstream with the bits flipped for every 32-bit word of the plaintext indeed revealed a burst of 4 bits, followed by a comparatively large dead time.

Using this model, it is also relatively straightforward to explain, why only the 3rd word in a 128-bit *configuration data* block was active. As detailed bitstream analysis shows, the reason for this is simply that the 32-bit commands preceding the "FDRI" configuration data block (see Fig. 6.) are not a multiple of four. As these commands are also encrypted into the ciphertext, the configuration data start is not aligned to the AES blocks. The "active" word visible in EOFM is therefore, simply always the last 32-bit word in an AES block. This hypothesis is supported by a simple check performed by shifting the configuration data in the bitstream by inserting 32-bit "no operation" (NOP) commands into the "header commands" block, see Fig. 6. As could be expected, a shift by one 32-bit command led to the next 32-bit word in the configuration data to become "active" as this also shifts the data in the AES blocks.

If this model is correct, the plaintext bitstream data can be extracted by probing all 32 bit lines of the AES output bus. However, as the relation between the spatial activity locations and the bit numbers is unknown, a bit location mapping would first need to be performed.

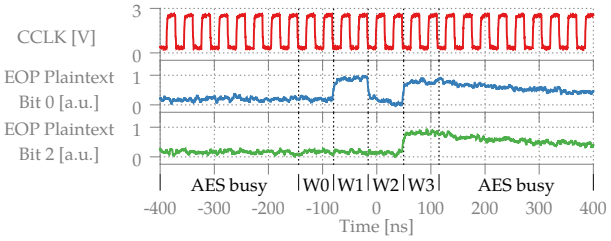


Figure 15: Optically extracted plaintext data for two bus lines of the 32-bit plaintext data bus as well as the CCLK signal. The plaintext bitstream data was 0101 for bit 0 and 0001 for bit 2. W0 to W3 denote the "data valid" time slots for word 0 to word 3 on the plaintext bus.

5.5 Logical to Spatial Mapping

If the model developed in the previous section is correct, the mapping between logical bus lines and their physical probing locations can be performed in a straightforward way: As all bits of the last 32-bit word in an AES block appear in the EOFM measurements, the location of a specific bus line can be found by only generating a fundamental frequency on the bit corresponding to that bus line. In other words, one specific bit in the "active" word is flipped for every AES block, while all other bits are set to "0". A complete mapping of the data bus would therefore in total require 32 EOFM measurements of the AES area, each performed with a bitstream only generating the desired frequency on a single bus line. The result of such a mapping for the AES output port can be seen in Fig. 14a. The figure shows the identified EOFM activity locations for each of the 32 bus lines. As multiple locations were active for each bus line, these have been denoted by a trailing “A” or “B”. For orientation, the locations have been overlayed onto an EOFM image were all bits were active. As can be seen, most isolated spots are actually composed of two smaller spots which stem from two different bus lines. This might raise concerns about the mixing of data of individual bus lines during later electro-optical probing. However, since there is a large enough gap to the neighboring spots, the beam can simply be parked at the respective edges of the composed activity spot during probing. Later probing tests actually showed that there is no bus line data mixing if this scheme is employed. Additionally, for every bus line, there are multiple probing locations available. During the mapping measurements on the AES output port, it became evident that there are also potential probing locations inside the AES logic. Fig. 14b shows the resulting mapping for activity spots in the AES logic next to the output port. This suggests that even if there was bus line data mixing at a certain probing location in the output port, the gates inside the logic could be used as an alternative. Further measurements even revealed additional potentially data carrying gates in the main core, which might be used as added alternatives.

As the bit mapping was now known with even multiple probing locations available, data extraction could be evaluated. For this, electro-optical probing was performed on the individual bus line

of the AES output port and the data compared to the plaintext bitstream data. As it turned out, all bus lines carried the expected data. Furthermore, it could be seen that the header commands, as well as configuration data, passes through these logic gates. This indicates that actually the entire plaintext bitstream is transmitted through this location and not just the "FDRI" configuration data block, see Fig. 6. Fig. 15 shows exemplary EOP probing waveforms for two of the 32 bus lines acquired with 5000 averagings. To synchronize the acquisition hardware, the trigger was set to "n-th edge" type on the configuration clock signal CCLK and was armed with the reconfiguration trigger signal PROGRAM_B. In this figure, four bits can be seen on each bus line, framed by the time where the AES is busy. The four bits belong to four words (W0-W3) of the plaintext bitstream data respectively. It can also be seen that the data on the bus lines is in phase with the externally available CCLK signal. It is evident that the complete bitstream data can be extracted by probing all bus lines in this way.

5.6 Expenditure of Time

As we assumed that the attacker has to rent the failure analysis equipment, we have tracked the time that was required to achieve each milestone. Tab. 1 gives an overview of these time expenditures up to the point were the plaintext gates were found and verified by EOP. We have included two time expenditure measures in this table. The first is the total time that the FA equipment was turned on, which also includes periods were the PC was locked or in standby. This is also the time that an attacker would have to pay for if she had rented the equipment. We also included a "usage time" in the table. Usage time is the sum of hours that the attacker has actively used software on the PC. It includes using the microscope control software as well as copying files in Windows Explorer and similar actions. This represents a best case scenario which the attacker could have achieved by working as fast as possible. These timings also include all overhead and wrong ways taken during our experiments. Noticeable are the timings for configuration logic localization and input bus width determination as they required a greater amount of time than the other steps. In the case of the "configuration logic" step, this is simply because it includes the initial setup and getting used to the board and equipment. The "input bus width" step is longer, as we invested some considerable effort into finding gates under the assumption of serial bitstream data in the device. To develop the complete attack, the logical-to-spatial mapping of Fig. 14 is also needed, which took two hours of measurement. Therefore, the total rent time for developing the attack would be 74.9 hours. As the setup used in this work can be rented for about 300 $/hour the approximate rent cost to develop the attack would be 22.5 k$.

6 DISCUSSION

6.1 Full Bitstream Extraction

Naturally, the results from Sect. 5 raise the question of how long a full bitstream extraction would take. In the case of the measurement depicted in Fig. 15, the device was configured to use the “33 MHz”

Table 1: Time spent on the FA microscope for each milestone of attack development. Usage time is the sum of hours that the attacker has actively used software on the PC.

Milestone	Powered On [h]	Usage Time [h]
Configuration Logic Localized	27.0	19.9
AES Logic Localized	9.1	8.0
Input Bus Width Determined	19.0	14.6
AES Output Localized	7.3	6.6
Successful Plaintext Probing	10.5	9.9
	Sum Powered On: 72.9	**Sum Usage: 58.9**

internal clock setting, no bitstream compression and a 1-bit wide SPI bus. If we take these settings as an example, one configuration cycle takes about 800 ms. In additional probing measurements, we determined that 100 averages is the lower limit to easily distinguish the bit states of the plaintext bus waveform without any filtering. EOP signal acquisition for one bus line would, therefore, require 80 s of averaging time. Multiplied by the 32-bit bus width, this amounts to 2560 seconds or 43 minutes of raw acquisition time for the whole bitstream. Note that this value would shrink accordingly if designs with a faster configuration clock, wider bitstream input bus or compression were used. Hence, we estimate that full bitstream extraction would take from a few hours to a few days of lab work, depending on the specific equipment and the overhead for setting up the experiment.

6.2 Stability of the Internal Clock

The approach we have taken in this work constitutes the worst case scenario regarding clock stability. Usually, it is recommended to use an external oscillator for designs, where a fast configuration time is desired [28]. In our experiments, however, we configured the FPGA to generate the clock internally. As the internal oscillator is not a stable high-quality clock source, this could introduce problems with averaging. Nevertheless, as the externally available configuration clock signal CCLK is in phase with the internal AES output, see Fig. 15, we were able to synchronize the acquisition equipment using this signal. We expect that a simpler trigger scheme could be utilized if a stable external oscillator is employed.

6.3 Optical Probing Availability

Optical probing systems are common FA equipment. Therefore, they can be rented at FA labs around the world, such as Presto Engineering [19] and Inscope Labs [8]. The system used for this work is also available for rent at about 300 $/h including operator cost.

6.4 Technology Size and Optical Resolution

Our results underline that it makes no sense to compare technology size of transistors to optical resolution. Such a comparison might be meaningful in failure analysis since it might be required

to resolve minimum size single transistors to find the cause of a failure. However, the optical resolution can be more relaxed in the case of security analysis of an IC. The best resolving lens in our experiments has a resolution of about 1 µm. Comparing this resolution to the device's technology size of 28 nm (or 0.028 µm) suggests that an optical attack is utterly futile. The results, however, demonstrate that an attack is not only possible but it can be developed in a few weeks. The real limiting factor for an attacker is not the technology size, but the distance of a probing location of interest to the next location carrying an interfering signal, compare Fig. 14. If the interfering signal and the signal of interest are close enough, they will both be illuminated by the focused light spot. Thus, they will both modulate the reflected light, and therefore, the resulting EOP signal will be a mix of both signals. If the interfering signal is uncorrelated to the signal of interest, the attacker can still mitigate this by averaging more waveforms, as the interfering signal behaves like an added noise source. On the other hand, if the mixing occurs between, for example, two or more bits of the plaintext data, the attacker will only be able to tell how many bits are set, and not which ones. As can be seen in Fig. 14, the separation between locations carrying different streams of data can actually be much larger than the technology size. Thus, when assessing potential optical attacks, the attack system's resolution should be compared to this "data separation pitch" and not to the technology size.

6.5 Device Damage and AES Key Loss

For other classes of attacks with similar capabilities, such as Focused Ion Beam (FIB) editing [16], key loss and device damage constitute a major concern during an attack. For example, while milling and polishing the device, power to the internal BBRAM key memory might be disrupted, and consequently, an attack rendered impossible. Even with successful preparation, if the chance of device damage for a single FIB edit is just 5 %, tapping into the plaintext bus would only have a $0.95^{32} = 19$ % chance of success. However, in the approach discussed in this work, there is no modification made to the FPGA. The only requirement is the addition of one coaxial cable to the CCLK signal on the printed circuit board and some means of triggering configuration, either through PROGRAM_B, a reset or a simple power-on loop. Therefore, loss of the BBRAM AES key during attack preparation is very unlikely. As the wavelength used for optical probing is larger than the silicon bandgap, no photocurrent is generated, and no disturbance in device operation is expected. As a matter of fact, we have exposed the device to full power light radiation for many hours without noticing any permanent or temporary effects during our experiments. In other words, a potential attacker has a virtually unlimited amount of time to explore and probe the device without worrying about damage.

7 POTENTIAL COUNTERMEASURES

There are three requirements for performing extraction of bitstream data as described in this work. The first is optical access to the transistors of the device. The second is the discoverability of the plaintext transistors through EOFM. The third is the availability of a trigger signal for the final EOP acquisition. All requirements can be targeted to hinder attacks.

7.1 Optical Access

There are two ways to protect against optical access, the first is detection and the second is prevention.

Silicon light sensors are conventional solutions, which have been proposed to detect the photons of the light beam. However, if the incident light has a longer wavelength than the silicon band gap, as in our experiments, the light sensors are only stimulated thermally. In this case, no electron-hole pairs are generated, and hence, a silicon photo sensor is not triggered. Thus, silicon light sensors can not be used to detect optical probing attacks with 1.3 µm light sources.

However, there is thermal stimulation during EOP/EOFM attempts, which can lead to immediate local disturbances in temperature and current of the transistors on the chip. This is often also referred to as Thermal Laser Stimulation (TLS) in the literature. Temperature and current variations influence the signal propagation delays of timing-dependent circuits, such as ring-oscillator Physically Unclonable Functions (PUFs). As proposed in [22], one potential countermeasure could be utilizing a ring-oscillator PUF and distributing its ring-oscillators close to the decryption core and the bus to detect optical probing attempts. In this case, any EOFM/EOP attempt would affect the behavior of the PUF with a high probability. The experimental results in [22] demonstrated that the attack can consequently be detected. As an anti-tamper reaction, the BBRAM key storage could be zeroized, or the whole decryption core could be locked down. Alternatively, less extreme countermeasure could be employed, like generating gibberish bitstream data, while optical probing is detected and returning to normal operation afterward. Since PUFs have been already considered for secret key generation inside FPGAs [17], it is conceivable to use the same PUFs as sensors to detect optical probing attacks as well. An additional advantage of this protection scheme is that it can also be employed by FPGA users on the reconfigurable logic without the need for hardware modifications.

A more complete approach would be to fundamentally prevent optical access to the chip. For the silicon frontside this is already a given because of the numerous metal layers of modern technologies. High-security ICs even implement active shield structures on the frontmost metal layers for protection. However, the silicon backside is currently lacking such protection, and worse, modern flip-chip packages even facilitate access.

An apparent solution would be to employ the same laser engraving process that is used for the device markings to decrease the substrate's optical quality on top of the AES core. However, although the signal quality and resolution are reduced, we could acquire EOFM signals from structures directly under device markings during our experiments. Additionally, the presence of this passive structure can not be monitored, and therefore, it could be removed by simple polishing. Nevertheless, such an approach would increase

the effort needed for an attack and the probability of device damage and key loss.

A better approach would be to add an entirely opaque layer to the backside of the chip. Passive physical protections, such as heat spreader lids, can already make physical access to the substrate more challenging. However, they are not designed with protection in mind, and thus, can also be removed without consequences [11]. Hence, a proper protection layer would need to be actively monitored. Such a scheme could make use of interactions between the protection structure and transistors on the chip to detect removal. To achieve this, first experimental results of a concept which uses special layers coated on the silicon substrate were presented in [2, 3]. In this case, the p-n junctions of standard transistors were operated in a way which causes them to emit photons. These photons then travel inside the silicon substrate to the backside, where a multilayer coating reflects them. This reflected light then travels back to other transistors on the chip, which are configured as light detectors. If an attacker removes the coating layers from the silicon substrate, the reflection characteristics of the backside are changed, and thus, attack attempts can be detected. As the layer is engineered to have non-standard angular-dependent reflection characteristics, simply coating the device with a new layer will not allow normal operation. Additionally, since standard transistors already present in the device are used as emitters and detectors, the overhead for the protection circuit is minimized. If such a scheme was implemented by FPGA vendors, it could effectively hinder all backside attacks including optical probing.

Therefore, there are experimentally validated concepts available to either detect [22] or completely prevent [2, 3] optical probing access. If further research is able to validate their mass production compatibility, they could protect future generations of ICs against optical backside attacks.

7.2 EOFM Discoverability

To increase the effort for the identification of the plaintext transistors through EOFM, designers of ASIC decryption cores could distribute and obfuscate the gates carrying the plaintext signal. However, it should be kept in mind, that this might add timing issues and overhead to the circuit.

Similar to power side-channel analysis countermeasures, adding gates carrying an inverted signal for cancellation of the EOP signal is also imaginable. Yet, this would require development and verification of suitable structures and ASIC design tools and is therefore not an out-of-the-box solution.

7.3 EOP Trigger

As the attacker needs a suitable trigger for EOP waveform acquisition, restricting access to the configuration clock signal CCLK is a potential countermeasure. An easy way for FPGA users to hinder an attacker could be the use of BGA packages for both FPGA and NVM and routing of CCLK and other relevant signals through the internal layers of the PCB. An attacker would then have to gain access through milling or drilling the PCB which increases the probability of damage or key loss. In this context, it might also be advantageous to use the key storage power supply as an additional makeshift protection structure. If the corresponding trace is routed on internal PCB layers above and below the CCLK signal this further increases the chance of key loss when milling.

It should be noted that this approach is only beneficial if an unstable clock source is used (see Sect. 6.2). Otherwise, the attacker can simply synchronize her equipment to a reset or power-on signal if these are freely available. An FPGA user still could circumvent this by unlinking power-on/reset and configuration start. This is achievable by using a two-stage configuration process in FPGAs which have dynamic or partial reconfiguration capability. If the first stage adds a random delay, it destroys the fixed relation between power-on/reset and configuration start.

A more thorough solution could come from the FPGA vendor side in the form of a circuit that destroys the fixed relationship between internal AES and external clock/data signals. Examples of circuits are small random delay FIFO buffers, the addition of jitter to the external signals or even a completely asynchronous internal clock. It should, however, be kept in mind that such a design might be challenging and also results in lower configuration speeds because of the added delays. Yet, if because of such an approach there is no direct or indirect external trigger available for the EOP acquisition, bitstream extraction can not take place.

8 CONCLUSION

In this work, we assessed the threat of conducting an optical probing attack against a commercial device with little or no knowledge about the security circuits available to the adversary. To this end, we mounted an attack against the bitstream encryption feature of a modern FPGA. While we had no prior knowledge about the underlying decryption cores, we demonstrated that renting the necessary equipment from a failure analysis lab for less than 10 days is enough for an attacker to localize the security circuits and extract plaintext data from the chip. Since modern silicon dies come in flip-chip packages, no chip preparation and silicon thinning were required for optical contactless probing in contrast to previous attacks. We also demonstrated that 1 µm optical resolution is enough to successfully attack a 28 nm technology device. The foremost reason that modern chips are vulnerable to optical probing is the lack of an effective protection of their backside. Hence, we provided a set of countermeasures, which can be integrated into modern ICs in different phases of manufacturing and application design, to protect them from this class of attack.

REFERENCES

[1] Amazon. 2017. Amazon Web Services, Inc.. https://aws.amazon.com/ec2/instance-types/f1/ [accessed 19 May 2017].

[2] Elham Amini, Ruslan Muydinov, Bernd Szyszka, and Christian Boit. 2017. Backside Protection Structure for Security Sensitive ICs. In *43rd International Symposium for Testing and Failure Analysis (November 5-9, 2017)*. Asm.

[3] Christian Boit, Shahin Tajik, Philipp Scholz, Elham Amini, Anne Beyreuther, Heiko Lohrke, and Jean-Pierre Seifert. 2016. From IC Debug to Hardware Security Risk: The Power of Backside Access and Optical Interaction. In *Physical and Failure Analysis of Integrated Circuits (IPFA), 2016 IEEE 23rd International Symposium on the*. IEEE, 365–369.

[4] Python Software Foundation. 2017. pycrypto 2.6.1. https://pypi.python.org/pypi/pycrypto [accessed 19 May 2017].

[5] Yohei Hori, Toshihiro Katashita, Akihiko Sasaki, and Akashi Satoh. 2012. Electromagnetic Side-channel Attack against 28-nm FPGA Device. *Pre-proceedings of WISA* (2012).

[6] Numato Lab. 2017. Skoll - Kintex 7 FPGA Development Board. https://numato.com/skoll-kintex-7-fpga-development-board/ [accessed 19 May 2017].

[7] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. 2016. No Place to Hide: Contactless Probing of Secret Data on FPGAs. In *Cryptographic Hardware and Embedded Systems–CHES 2016*. Springer, 147–168.

[8] Inscope Labs PTE LTD. 2017. http://www.inscopelabs.com [accessed 19 May 2017]. (2017).

[9] William Luis, G Richard Newell, and Kenneth Alexander. 2015. Differential Power Analysis Countermeasures for the Configuration of SRAM FPGAs. In *Military Communications Conference, MILCOM 2015-2015 IEEE*. IEEE, 1276–1283.

[10] Cade Metz. 2016. Microsoft Bets Its Future on a Reprogrammable Computer Chip. https://www.wired.com/2016/09/microsoft-bets-future-chip-reprogram-fly/ [accessed 19 May 2017]. Wired.

[11] Amir Moradi. 2013. Altera vs. Xilinx: Which One Keeps Your Design Hidden?. https://www.emsec.rub.de/media/attachments/files/2016/05/AmirTalk_2013-08-22-CHES-Rump.pdf]. *Rump Session CHES 2013*.

[12] Amir Moradi, Alessandro Barenghi, Timo Kasper, and Christof Paar. 2011. On the Vulnerability of FPGA Bitstream Encryption against Power Analysis Attacks: Extracting Keys from Xilinx Virtex-II FPGAs. In *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 111–124.

[13] Amir Moradi, Markus Kasper, and Christof Paar. 2012. Black-Box Side-channel Attacks Highlight the Importance of Countermeasures. *Topics in Cryptology–CT-RSA 2012*, 1–18.

[14] Amir Moradi, David Oswald, Christof Paar, and Pawel Swierczynski. 2013. Side-Channel Attacks on the Bitstream Encryption Mechanism of Altera Stratix II: Facilitating Black-Box Analysis using Software Reverse-Engineering. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*. ACM, 91–100.

[15] Amir Moradi and Tobias Schneider. 2016. Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series. In *Constructive Side-Channel Analysis and Secure Design – COSADE 2016*. Springer.

[16] Dmitry Nedospasov, Jean-Pierre Seifert, Clemens Helfmeier, and Christian Boit. 2013. Invasive PUF Analysis. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*. IEEE, 30–38.

[17] Ed Peterson. 2015. White Paper WP468: Leveraging Asymmetric Authentication to Enhance Security-Critical Applications Using Zynq-7000 All Programmable SoCs. http://www.xilinx.com/support/documentation/white_papers/wp468_asym-auth-zynq-7000.pdf [accessed 21 Jan 2016]. *Xilinx, Inc. San Jose, CA* (2015).

[18] Stephan Preibisch, Stephan Saalfeld, and Pavel Tomancak. 2009. Globally Optimal Stitching of Tiled 3D Microscopic Image Acquisitions. *Bioinformatics* 25, 11, 1463–1465.

[19] Presto Engineering, Inc. 2017. http://www.presto-eng.com [accessed 19 May 2017]. (2017).

[20] Sergei Skorobogatov and Christopher Woods. 2012. Breakthrough Silicon Scanning Discovers Backdoor in Military Chip. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 23–40.

[21] Pawel Swierczynski, Amir Moradi, David Oswald, and Christof Paar. 2015. Physical Security Evaluation of the Bitstream Encryption mechanism of Altera Stratix II and Stratix III FPGAs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 7, 4, 34.

[22] Shahin Tajik, Julian Fietkau, Heiko Lohrke, Jean-Pierre Seifert, and Christian Boit. 2017. PUFMon: Security Monitoring of FPGAs using Physically Unclonable Functions. *On-Line Testing Symposium (IOLTS), 2017 IEEE 23rt International*.

[23] Mark M. Tehranipoor, Ujjwal Guin, and Swarup Bhunia. 2017. Invasion of the Hardware Snatchers: Cloned Electronics Pollute the Market. *IEEE Spectrum*.

[24] Stephen M Trimberger and Jason J Moore. 2014. FPGA Security: Motivations, Features, and Applications. *Proc. IEEE* 102, 8 (2014), 1248–1265.

[25] John Villasenor and Mohammad Tehranipoor. 2013. The Hidden Dangers of Chop-Shop Electronics: Clever Counterfeiters Sell Old Components as New Threatening both Military and Commercial Systems. *IEEE Spectrum (cover story)*.

[26] Eric W Weisstein. 2004. Fourier transform. *Wolfram Research, Inc.*.

[27] Kyle Wilkinson. 2015. Using Encryption to Secure a 7 Series FPGA Bitstream. https://www.xilinx.com/support/documentation/application_notes/xapp1239-fpga-bitstream-encryption.pdf [accessed 19 May 2017]. *Xilinx, Inc. San Jose, CA*.

[28] Xilinx. 2016. 7 Series FPGAs Configuration User Guide. https://www.xilinx.com/support/documentation/user_guides/ug470_7Series_Config.pdf [accessed 19 May 2017]. *Xilinx, Inc. San Jose, CA*.

[29] Xilinx. 2016. 7 Series FPGAs GTX/GTH Transceivers. https://www.xilinx.com/support/documentation/user_guides/ug476_7Series_Transceivers.pdf [accessed 19 May 2017]. *Xilinx, Inc. San Jose, CA*.

[30] Arthur Yang. 2016. Using SPI Flash with 7 Series FPGAs. https://www.xilinx.com/support/documentation/application_notes/xapp586-spi-flash.pdf [accessed 19 May 2017]. *Xilinx, Inc. San Jose, CA*.