# PUFMon: Security Monitoring of FPGAs using Physically Unclonable Functions

**Conference Paper** · July 2017

CITATIONS
0

READS
177

**5 authors**, including:

Shahin Tajik
Technische Universität Berlin
**22** PUBLICATIONS **86** CITATIONS
SEE PROFILE

Jean-Pierre Seifert
Technische Universität Berlin
**108** PUBLICATIONS **1,201** CITATIONS
SEE PROFILE

# PUFMon: Security Monitoring of FPGAs using Physically Unclonable Functions

Shahin Tajik[1], Julian Fietkau[1], Heiko Lohrke[2], Jean-Pierre Seifert[1], and Christian Boit[2]

[1] Security in Telecommunications, Dept. of Software Eng. and Theoretical Computer Science
[2]Semiconductor Devices, Dept. of High-Frequency and Semiconductor System Tech.
Technische Universität Berlin, Berlin, Germany
{stajik, jfietkau, jpseifert}@sec.t-labs.tu-berlin.de
{lohrke}@mailbox.tu-berlin.de, {christian.boit}@tu-berlin.de

***Abstract*—Mainstream FPGAs and programmable SoCs employ different countermeasures during configuration and runtime to mitigate physical attacks. However, it has been demonstrated that sophisticated active attack techniques, such as laser voltage probing, can still bypass the bitstream protections during the configuration phase. On the other hand, although the security monitoring IP cores provided by FPGA vendors can ensure the physical security during the runtime of applications, they are unable to detect such attacks during configuration. In this work, we propose a novel approach to using PUFs as physical sensors to monitor the integrity of FPGAs against active attacks. Small modifications in existing PUF architectures enable us to design a PUF-based security scheme, which can be deployed for integrity monitoring and authentication/key generation at the same time. We evaluate the effectiveness of our framework against a range of powerful attacks, such as optical probing and fault attacks. We further discuss how this scheme can be deployed during bitstream configuration in FPGAs with partial reconfiguration capability.**

***Index Terms*—Anti-Tamper; FPGA and SoC Security; Laser Voltage Probing; Physically Unclonable Functions;**

## I. Introduction

Security features of Field Programmable Gate Arrays (FPGAs) and programmable System on Chips (SoCs) have evolved over time to cope with physical attacks. For instance, using asymmetric authentication, key rolling and side-channel resistant decryptors on the most recent FPGA models assures the confidentiality and integrity of the bitstream against side-channel attacks [1], [2]. Furthermore, deploying Physically Unclonable Function (PUF) as a secure key storage method prevents the direct readout of secret keys by semi- and fully-invasive techniques [1]. On the other hand, FPGA vendors provide customers with proprietary soft security monitoring intellectual properties (IPs) to guarantee a post-configuration failsafe security scheme. Xilinx SecMon [3] and Microsemi EnforcIT [4] are examples of such security monitoring IPs, which are configured along with the bitstream on the device. These IPs utilize the existing sensors inside the FPGAs to monitor the integrity of the device during runtime.



Recently, a new attack [5] against FPGAs relying on Laser Voltage Probing (LVP) has been introduced, which can bypass integrated configuration protections. This technique enables the attacker to probe volatile and on-die-only secret data on FPGAs in a contactless way from the IC backside. The first and foremost reason, which makes this attack feasible is the lack of physical protection for the FPGA backside. Even the security monitoring IPs cannot be used to detect such attacks during configuration since they are included in the bitstream, and they are activated only when the FPGA is totally configured. Moreover, other classes of the active attacks (e.g., fault attacks) are major threats to earlier generation FPGAs, which are still largely deployed in the field and do not support the new security schemes.

One similarity among different active attacks is their instant disturbance on a set of physical parameters, e.g., temperature and current. In the case of LVP, the incoming laser photons increase the temperature of the wires and transistors locally on the chip. Fault attacks, such as clock manipulation and reconfiguration attacks cause current variations on the die. The local temperature and current variations can affect the propagation delays of the electrical signals in the timing-dependent circuits, such as delay-based PUFs [6]. Since soft PUFs (i.e., PUF configurations for the FPGA fabric) have been already considered as a solution for key generation inside the FPGAs [1], it is tempting to use the same PUFs as physical sensors to detect the active attacks as well.

**Our Contribution.** The main contribution of this paper is the proposal of a PUF-based security monitoring scheme (PUFMon) to detect active attacks. We demonstrate that a small modification to an existing PUF architecture [7] enables us in a certain degree to detect the induced local temperature and current variations by active attacks. We assess the usefulness of the PUFMon by conducting optical contactless probing, clock manipulation and reconfiguration attacks against our Proof-of-Concept implementations on Altera FPGAs. Finally, our experiments show that despite attack detection, the challenge-response pairs (CRP) behavior of the PUF remains stable for authentication and key generation purposes.

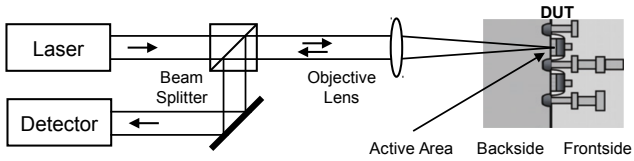


Fig. 1: Simplified illustration of LVP signal acquisition [5].

## II. Background

### A. Laser Voltage Probing

Laser Voltage Probing (LVP) and Laser Voltage Imaging (LVI) techniques have been introduced in the field of failure analysis to debug the nanoscale transistors on the chip. While LVP can be used to directly probe the electrical signals on the transistors through the silicon backside, LVI can be used to create an activity map of active circuits. In both cases, the laser photons pass through the silicon substrate on the IC backside to reach the transistors, which leads to a partial absorption and a partial reflection of the laser beam. In the case of LVP, 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. Therefore, the data passing through a node can be probed. For performing the LVI on the other hand, the reflected light is fed to a detector with a narrow band frequency filter while the laser scans the device. As a result, the nodes with a switching frequency equal to the frequency filter can be detected and localized on the chip. In this paper, we refer to both techniques as LVP. The LVP technique has been used in the security-related literature [5] to attack PUF-based configuration schemes to probe the generated secret keys directly.

### B. Fault Attacks against FPGAs

Fault attacks against FPGAs can result in transient and semi-permanent faults. Clock, voltage and temperature manipulations during the runtime are the classical fault attacks, which cause transient faults. On the contrary, flipping the values stored in the configuration memories of lookup tables and switch boxes generates semi-permanent faults and leads to the reconfiguration of FPGAs. In other words, the faults are permanent as long as the FPGA is powered on or not rebooted. Reconfiguration attacks can be launched by tampering with the bitstream [8] and voltage glitching [9] during the configuration phase or laser fault injection [10] during the runtime.

A set of transient faults can be detected by the vendor's security monitoring IPs using dedicated temperature, voltage and user clock sensors implemented inside modern FPGAs. Reconfiguration attacks can be detected by applying asymmetric authentication schemes and carrying out an integrity check of configuration cells during configuration phase and runtime, respectively. However, these schemes can be only applied to recent generations of FPGAs, and earlier generations and less expensive FPGAs do not support them.

### C. Related Work

Although one of the initial promises of PUFs was its tamper-evidence feature [11], the sensitivity of PUFs to environmental variations is usually not desired for key storage and generation purposes. Therefore, the idea of using PUFs as monitoring sensor has been largely neglected in the past and a great deal of attention has been paid to the design of PUFs, whose CRPs are stable under different physical conditions. Recently, the concept of PUFs has been considered as a sensor proving the existence of a physical condition to a verifier [12]. Nevertheless, timing-based hardware primitives, similar to delay-based PUFs, have earlier been used to detect fault attacks or hardware Trojans on the chip. For instance, a countermeasure based on a single RO connected to a PLL inside an FPGA was proposed, which is able to detect laser, and EM fault injection attacks [13]. In another attempt, an RO that operates over coupled lines in a differential mode has been used on ASICs to detect the probing attempts [14].

## III. Methodology

### A. Requirements for Detecting an Attack

Active attacks, such as LVP and fault attacks in general, induce temperature and current variations into the chip. These local variations change the signal propagation delays of the transistors. Naturally, an ideal sensor should have a high resolution to measure tiny variations in the different physical parameters. However, to detect the full range of active attacks, multiple conditions have to be fulfilled.

*1) Large Spatial Coverage:* Although most active attacks have a global effect on the target FPGA, performing LVP over a certain area of the chip only locally increases the temperature of the wires and transistors. Therefore, an ideal sensor should cover the whole area of the chip to detect local variations.

*2) Large Temporal Coverage:* Active attacks can be performed using only a very short amount of time. The irradiating of individual gates or registers during either an LVP attempt or glitching clock/voltage signals is potentially a very fast process. Hence, the sensor should continuously monitor the physical conditions of the chip to successfully detect an attack.

*3) Security:* The attacker might make an effort to deactivate the sensor or tamper with the sensor output to hide her attack attempts. In an ideal case, any physical modification of the sensor should lead to an irreversible damage to the sensor.

### B. Sensor Candidates

The behavior of timing-based circuits such as RO networks (RONs) [15] and delay-based PUFs [6] heavily rely on the propagation delays of their composing combinatorial components. To have a large spatial coverage, the ROs in a RON with virtually equal frequencies can be distributed all over the FPGA, see Fig. 2a. Therefore, performing local attacks (e.g., LVP) will slightly shift the frequencies of ROs, which are in, or close to the probed area. The frequency deviation of affected ROs can be compared to the mean frequency of all ROs to detect the attack. Similarly, the individual ROs of RO PUFs can be realized in a distributed way on the chip.

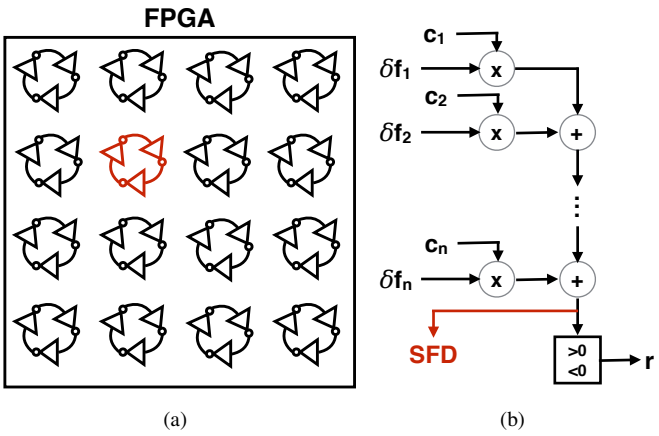


Fig. 2: (a) A distributed placement of the ROs inside the FPGA. Performing LVP can shift slightly the frequency of an RO, which is in, or close to the probed area. (b) The modified architecture of the RO sum PUF. The summation of frequency differences is measured directly before making a decision about the output of the PUF.

Other delay-based PUFs (e.g., arbiter PUFs) can be employed inside an FPGA in a similar manner to cover the whole area of the chip. For instance, the multiplexers or the inverters of a large Arbiter PUF can be placed and distributed manually all over the FPGA fabric.

However, satisfying the temporal coverage requirements, especially for PUF-based sensors, is more challenging. For instance, although the laser irradiation can instantly alter the delays of one or more PUF components, the affected components might not be active during the attack period. If we assume that an Arbiter PUF is used as a sensor, by enabling the PUF for a specific challenge, each stage of the Arbiter PUF is active only for a few picoseconds to pass the incoming signal from the previous stage to the next one. In this case, instant delay alteration of one stage prior or after the signal handover will not have any effect on the behavior of the PUF. Hence, a sensor with constantly active elements should be selected if a high temporal coverage is required. Among digital intrinsic hardware primitives, RO sum PUFs [7] and RONs can, therefore, offer better temporal coverage since their ROs can always be made active to sense the anomalies.

A permanent physical modification of both RONs and delay-based PUFs with a high probability leads to the destruction of them. Moreover, their deactivation leads to altered outputs. Thus, they can be considered tamper-evident and satisfy the security condition.

### C. Combining RO Sum PUFs and RONs

RONs have been used to create signatures, detect small current variations as well as hardware Trojans in ASICs. Upon activation of the ROs, the counters count the number of rising edges at the outputs of the ROs within a predefined interval. The frequency of each RO is measured separately by a counter and summed up to be analyzed externally. Mounting active attacks can *directly* alter the frequency values, and the attack can be detected with a high probability. However, RONs are neither common IPs for FPGAs nor well suited for key generation and authentication simultaneously. The latter is due to the lack of both exponential input space and binary responses.

On the other hand, RO sum PUFs consist of $n$ pairs of ROs, whose outputs are connected to binary counters. Similar to RONs, after a predefined period, the states of the counters of two adjacent ROs are sampled. The sampled values are then subtracted from each other to generate $n$ frequency difference values. Based on an applied $n$-bit challenge to the PUF, each frequency difference value is multiplied by $+1$ or $-1$. Finally, all the values are summed, and the final response of the PUF is $0$ if the sum is negative. Otherwise it equals $1$. An RO sum PUF has an exponential CRP space, and therefore, it can be utilized for authentication and key generation purposes. However, it should be noted that changes in the frequencies of a small number of ROs that are under an active attack do not necessarily change the *binary* response of the PUF, which is undesired in the case of a PUF acting as an attack detection sensor.

To have an exponential CRP space and instant changes in the response during active attacks, the concepts of RON and RO sum PUF can be combined. If the summation of the frequency differences (SFD) in the RO sum PUF is measured before the decision making (similar to a RON), immediate changes in the behavior of ROs can be observed, see Fig. 2b. Meanwhile, if the binary responses of the PUF are not affected, the PUF can be utilized for key generation and authentication purposes. We refer to this PUF-based security monitoring scheme as PUFMon.

### D. Enrollment and Verification

Similar to PUF-based authentication, a PUF-based physical integrity sensor has to be evaluated in two phases. First, in the enrollment phase for each PUF, a set of CRPs is measured and stored in a database in a trusted field. Later in the verification phase, the enrolled challenges are retrieved from the database and applied to the PUF, and the outputs generated by the PUF are compared to the enrolled outputs. Since the SFDs are used later in the field to detect an attack, the SFDs of the PUFMon are measured several times in a normal and fault-free condition to obtain their maximum and minimum values. These values can be gathered in parallel with the actual binary responses of the RO sum PUF during the enrollment phase in a trusted field. Afterwards, in an adversarial field, an alarm can be raised if a predefined percentage of the SFDs does not lie within the min/max interval determined in this way.

There are different options for storing the SFD min/max limits of the PUF. One option is to transmit a set of enrolled limits encapsulated in the encrypted first stage boot loader (FSBL) or boot0 to the FPGA and store them in the volatile memories. This technique can only be applied to FPGAs with partial reconfiguration capability. In this scenario, the current SFDs of the PUFMon can be compared offline with the SFD

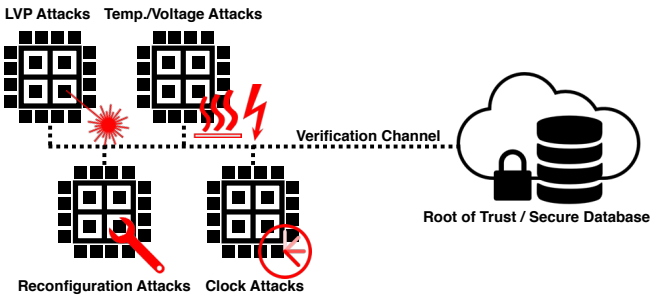


Fig. 3: Online verification of the SFDs with a RoT/secure database.

limits stored inside the chip to detect anomaly conditions. Although offline verification can be effective against non-invasive active attacks during runtime, the SFD limits might be vulnerable to tampering by invasive attacks. Another option is to store the SFD limits externally in a root of trust (RoT) or a secure database. In this case, the FPGA should communicate with the database online and verify the behavior of the FPGA, see Fig. 3. Finally, in a hybrid approach, while the PUFMon monitors the integrity of the FPGA offline, a server can occasionally update the stored challenges and SFD limits on the chip to raise the security level. It should be noted that in an ideal case the communication between FPGA and the database should be secured by encryption and authentication schemes. Otherwise, the attacker might be able to characterize the PUF by intercepting the SFDs.

## IV. Experimental Setup

### A. Device Under Test

The devices used for our experiments were Altera Cyclone IV FPGAs manufactured with a 60 nm process [16]. We utilized two different Altera Cyclone IV samples for semi-invasive and non-invasive tests. For optical probing and voltage glitching experiments, we used samples with the part number `EP4CE6E22C8N`. The 144 pin TQFP packages have been chosen in order to facilitate the sample preparation. First, we removed the exposed ground pad on the backside of the chip packages. Afterward, the samples were thinned by an Ultratec ASAP-1 polishing machine to a obtain a silicon thickness of 25 µm. The samples were soldered inversely to a custom designed printed circuit board (PCB). Finally, we reconnected the bond wires leading to the exposed ground pad to the ground on the board. To test the effects of clock manipulation and reconfiguration attacks, we used samples with the part number `EP4CE22F17C6N`. In this case, samples are available in the Terasic DE0-Nano products, which are common FPGA development boards.

### B. PoC FPGA Implementation

For a Proof-of-Concept, we implemented a 64-bit RO sum PUF consisting of 128 ROs on the FPGA of the Terasic development board. Furthermore, we implemented a 16-bit RO sum PUF comprised of 32 ROs on the FPGA used on the custom PCB. The changed size, in the latter case, is due to fewer resources being needed to cover this type of FPGA completely. Each RO in our design consists of 5 inverters. The ROs were placed manually and distributed all over the FPGA silicon area. In both cases, the RO network utilized about 15% of FPGA resources. The frequencies of ROs are measured by 16-bit counters. The challenges are transmitted from a laptop via the UART protocol to the FPGA and the generated SFDs and responses are sent back on the same channel.

### C. Measurement Setup

A Hamamatsu PHEMOS 1000 laser scanning microscope was used to measure the effect of a laser beam on the PUF. There are two different laser sources with the wavelengths 1064 nm or 1300 nm available on the system. The maximum power of the 1064 nm and 1300 nm lasers are 200 mW and 100 mW at the laser source, respectively. A PC running the PHEMOS software controls the optical setup.

## V. Results

In the enrollment phase of all experiments, we have selected a single set of 100 randomly chosen challenges, where each set is evaluate 50 times to gather their minimum and maximum SFDs. Later, in the monitoring phase, we have applied this set of challenges in a loop and compare them with results stored in an external database during the enrollment phase. During the attack period the number of out of bound SFDs (i.e., larger/less than the enrolled maximum/minimum values) has been registered. Finally, This data can then be used later to set a general statistical threshold to detect different attacks.

### A. Monitoring of LVP

We conducted our LVP experiments with two different wavelengths. During the monitoring phase, we performed LVP over an area of the chip from the IC backside. Each step of our experiments consists of 10 rounds of SFD evaluation by applying the set of 100 enrolled challenges. We started our first 10 rounds of monitoring rounds without performing LVP. Afterward, step by step we increased the power of the laser by 10% for 10 rounds. After each step, the laser was turned off for 10 rounds, see Fig. 4.

In order to interpret the experimental results, the effect of photons with different wavelengths on the FET transistors should be understood. The silicon substrate is more absorptive at 1064 nm than at 1300 nm, and thus, less 1064 nm photons reach the transistors. However, since the photons with the wavelength of 1300 nm contain less energy than the band gap of the silicon, they mainly have thermal effects. On the other hand, the photons with a wavelength of 1064 nm have higher energy than the band gap of the silicon, and hence, in addition to thermal effects they generate electron-hole pairs, which leads to current induction in the transistors. Stimulation of transistors with 1064 nm and 1300 nm photons is called Photoelectric Laser Stimulation (PLS) and Thermal Laser Stimulation (TLS), respectively [17]. As a result, the

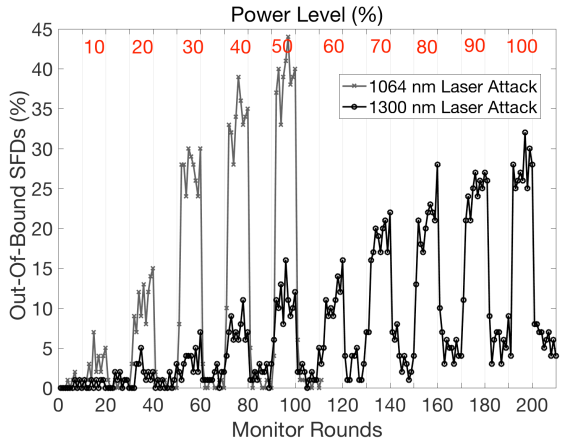


Fig. 4: The effects of the performing LVP/LVI with the 1064 nm and 1300 nm laser on the SFDs.

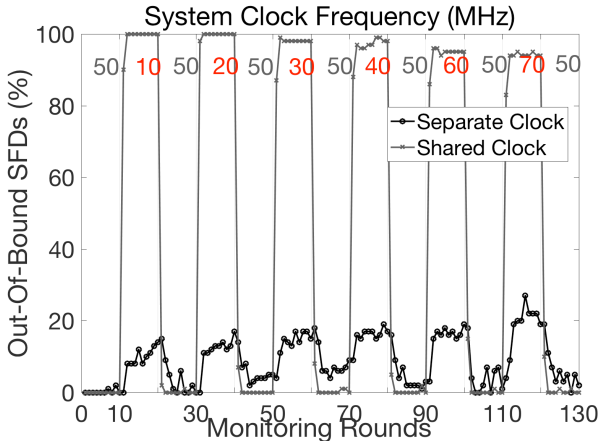


Fig. 5: The effects of the clock manipulation attack on the SFDs.

shift in frequency of the ROs is higher when using the 1064 nm laser, and the detection is more probable. As can be seen in Fig. 4, given enough laser power, there is detectable laser influence on SFDs. The maximum power for our 1064 nm laser experiment was 50% because there was a risk of damaging the transistors permanently.

A phenomenon that can be observed during high power experiments with the 1300 nm laser is the shift of the SFDs even after the laser is powered off. Naturally, increasing the laser power leads to an increase in the amount of the heat deposited in the chip during the *laser-on* period. Therefore, a few SFDs of the RO sum PUF still behaves out of bound at the end of the *laser-off* period.

### B. Monitoring of Clock Manipulations

In an RO PUF, a clock signal is required for sampling the states of the counters, which measure the frequencies of the ROs. Hence, any changes in the clock frequency can result in a shift in the time of sampling of the counters' values leading to variations in the SFDs. To assess clock manipulation detection, we implemented an AES-128 core on the FPGA, which could be the target of the clock tampering. The AES core is running in a loop during the clock manipulation. We set the target clock frequency to 50 MHz and manipulate it to values between 10 MHz and 70 MHz in 10 MHz steps. For each specific clock frequency, we have evaluated the SFDs for 10 rounds.

To assess the behavior of the PUF during a clock manipulation period, we considered two scenarios. In the first scenario, we realized the target clock line in a way that it drives both the PUF and AES circuits. In this case, manipulation of the clock frequency changed the sampling time of the counters in our PUF and also altered the characteristic voltage variations on the FPGA fabric caused by the AES core. Consequently, we can observe significant deviations of the SFDs from the min/max interval, see Fig. 5. In the second scenario, we realized two different clock sources and the target clock is only connected to the AES core, and the clock signal of PUF was not tampered during the attack. In contrast to the first scenario, the SFDs can be altered only due to variations caused by the changed behavior of the AES core. Hence, less SFD variation can be observed in comparison to the scenario, where a shared clock was tampered with, see Fig. 5.

### C. Monitoring of Reconfiguration Attacks

We emulated reconfiguration attacks by manually activating and deactivating particular hardware primitives, such as Linear-Feedback Shift Registers (LFSRs)-based PRNGs and RO-based TRNGs implemented in parallel with PUFMon. Deactivation of these elements have an impact on the power consumption of the chip. The LFSR in our design consisted of 64 registers and was running at 50 MHz. The RO contained 5 inverters and could be activated/deactivated by an AND gate. In two independent experiments, we ran the PRNG and TRNG circuits for 10 rounds. In the second phase, we deactivated them to simulate a reconfiguration attack. It was observed that the SFDs were changed accordingly, see Fig 6. Note that the amount of the out of bound SFDs for the LFSR and the RO cannot be compared since the impact on the power consumption is highly dependent on the target primitive and its placement on the chip.

## VI. Discussion

### A. Strengths of PUFMon

While the security monitoring IPs provided by FPGA vendors can be implemented only in few modern FPGA generations, PUFMon can be deployed on each and every generation of FPGA platforms, especially on less expensive FPGAs without security features commonly used in the field. Moreover, in order to configure these proprietary IPs on the FPGAs, access to the user bitstream is required by the vendors, which might be undesired for some applications. In contrast to this, PUFMon can be implemented independent of the user application. Finally, the behavior of the PUFMon is entangled with particular FPGA platforms. If the SFD limits of one

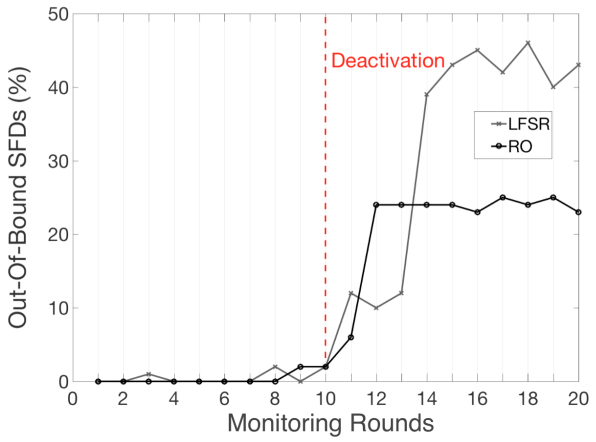


Fig. 6: The effects of deactivation of an RO and an LFSR on the SFDs.

device are divulged, the PUFMon behavior of other devices remains unknown.

To enable a successful key generation and authentication, the CRP behavior of the PUF should be stable. We have observed that despite large changes in the SFD values of the PUF under attack, most binary responses of the PUF remained intact. Accordingly, if our proposed modified RO sum PUF is considered as a soft PUF IP, it can be used for authentication, key generation and monitoring at the same time. In this case, the PUF configuration can be transmitted from a RoT in the FSBL to the FPGA. Afterward, the bitstream can be transmitted from the RoT and configured on the FPGA via partial reconfiguration. During configuration, PUFMon can communicate with the RoT to verify the correct functionality of the FPGA.

### B. Weaknesses of PUFMon

We have further conducted temperature and voltage manipulation attacks against our Proof-of-Concept implementation similar to the experiments in Sect. V. Based on our observations, the SFDs are sensitive to small variations in the power supply voltage and global temperature. While high sensitivity enables the detection of the voltage and clock glitching for a very short period, it can also raise false alarms. To solve this issue, one can set the detection threshold higher at the cost of a lower detection probability for more local attacks, such as LVP. The response of more stable PUFs architectures, such as BR PUF and Arbiter PUF, might be better indicators to detect temperature [12] and voltage manipulations at the cost of less temporal coverage.

Another weakness of the PUFMon is the high power consumption of its ROs, which makes it unattractive for low power applications. One could reduce the number of the ROs and place them only close to the critical IPs. However, the spatial coverage of the PUFMon will be decreased. Another solution is to activate the PUFMon only within the critical periods, such as during configuration and encryption/decryption phases. Besides, PUFMon could have standby times to reduce the overall power consumption. Nonetheless, in both cases, the temporal coverage of the PUFMon is decreased.

## VII. Conclusion

In this work, we have made a first attempt to evaluate the feasibility of using PUFs as an integrity monitoring sensor for FPGAs. We have evaluated the performance of our PUF-based sensor against different active attacks, namely LVP, clock manipulations and reconfiguration attacks. Despite the advantages of the PUFMon, its detection capability might not be comparable to proprietary solutions. Hence, PUFMon should be considered as an add-on to the existing security schemes to enhance the probability of the attack detections.

## References


[1] E. Peterson, "White Paper WP468: Leveraging Asymmetric Authentication to Enhance Security-Critical Applications Using Zynq-7000 All Programmable SoCs," *Xilinx, Inc. San Jose, CA*, 2015.

[2] W. Luis, G. Richard Newell, and K. Alexander, "Differential Power Analysis Countermeasures for the Configuration of SRAM FPGAs," in *Military Communications Conference, MILCOM 2015-2015 IEEE*. IEEE, 2015, pp. 1276–1283.

[3] Xilinx, "Security Monitor IP: Industry-Leading Programmable Device Security Protecting IP and Mission Critical Data," *Xilinx, Inc. San Jose, CA*, 2015.

[4] Microsemi, "EnforcIT Security Monitor," *Microsemi Corporation, Aliso Viejo, CA*, 2015.

[5] H. Lohrke, S. Tajik, C. Boit, and J.-P. Seifert, "No Place to Hide: Contactless Probing of Secret Data on FPGAs," in *Cryptographic Hardware and Embedded Systems–CHES 2016*. Springer, 2016, pp. 147–168.

[6] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications," in *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*. IEEE, 2004, pp. 176–179.

[7] M.-D. M. Yu and S. Devadas, "Recombination of Physical Unclonable Functions," *United States. Dept. of Defense*, 2010.

[8] P. Swierczynski, G. T. Becker, A. Moradi, and C. Paar, "Bitstream Fault Injections (BiFI)–Automated Fault Attacks against SRAM-based FPGAs," *IACR Cryptology ePrint Archive*, 2016.

[9] C. O'Flynn, "Fault Injection using Crowbars on Embedded Systems," *IACR Cryptology ePrint Archive*, 2016.

[10] H. Lohrke, P. Scholz, C. Boit, S. Tajik, and J.-P. Seifert, "Automated Detection of Fault Sensitive Locations for Reconfiguration Attacks on Programmable Logic," in *42nd International Symposium for Testing and Failure Analysis (November 6-10, 2016)*. ASM, 2016.

[11] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer Berlin Heidelberg, 2013.

[12] U. Rührmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson, "Virtual Proofs of Reality and their Physical Implementation," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 70–85.

[13] W. He, J. Breier, S. Bhasin, N. Miura, and M. Nagata, "Ring Oscillator under Laser: Potential of PLL based countermeasure Against Laser Fault Injection," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2016 Workshop on*. IEEE, 2016, pp. 102–113.

[14] S. Manich, M. S. Wamser, and G. Sigl, "Detection of probing attempts in secure ics," in *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 134–139.

[15] X. Zhang and M. Tehranipoor, "RON: An On-chip Ring Oscillator Network for Hardware Trojan Detection," in *2011 Design, Automation & Test in Europe*. IEEE, 2011, pp. 1–6.

[16] Altera, "Cyclone IV Device Handbook," *Altera Corporation, San Jose*, 2014.

[17] S. K. Brahma, J. Heinig, A. Glowacki, R. Leihkauf, and C. Boit, "Distinction of Photo-Electric and Thermal Effects in a MOSFET by 1064 nm Laser Stimulation," in *2006 13th International Symposium on the Physical and Failure Analysis of Integrated Circuits*. IEEE, 2006, pp. 333–339.