Securing Cloud FPGAs Against Power Side-Channel Attacks: A Case Study on Iterative AES

Nithyashankari Gummidipoondi Jayasankaran, *Student Member, IEEE,* Hao Guo, *Student Member, IEEE,* Satwik Patnaik, *Member, IEEE,* Jeyavijayan (JV) Rajendran, *Senior Member, IEEE,* and Jiang Hu, *Fellow, IEEE*

Abstract—The various benefits of multi-tenanting, such as higher device utilization and increased profit margin, intrigue the cloud field-programmable gate array (FPGA) servers to include multi-tenanting in their infrastructure. However, this property makes these servers vulnerable to power side-channel (PSC) attacks. Logic designs such as ring oscillator (RO) and time-to-digital converter (TDC) are used to measure the power consumed by security critical circuits, such as advanced encryption standard (AES). Firstly, the existing works require higher minimum traces for disclosure (MTD). Hence, in this work, we improve the sensitivity of the TDC-based sensors by manually placing the FPGA primitives inferring these sensors. This enhancement helps to determine the 128-bit AES key using 3.8K traces. Secondly, the existing defenses use ROs to defend against PSC attacks. However, cloud servers such as Amazon Web Services (AWS) block design with combinatorial loops. Hence, we propose a placement-based defense. We study the impact of (i) primitive-level placement on the AES design and (ii) additional logic that resides along with the AES on the correlation power analysis (CPA) attack results. Our results showcase that the AES along with filters and/or processors are sufficient to provide the same level or better security than the existing defenses.

Index Terms—PSC attack, CPA attack, cloud FPGA security, ring oscillators (ROs), time-to-digital converters (TDC), minimum traces for disclosure (MTD)

1 Introduction

1.1 Security Vulnerabilities in Cloud FPGAs

Hardware accelerators based on the field programmable gate arrays (FPGAs) support parallel processing and have better performance and power efficiency than CPUs and GPUs, respectively. Owing to these performance benefits, they are best suited for deploying compute-intensive tasks such as big data analytics, real-time video processing, and genomics research [1]. Hence, to meet user demands, cloud service providers such as Amazon Web Services (AWS), Microsoft Azure, IBM Cloud, and Texas Advanced Computing Center (TACC) have replaced CPUs with FPGAs in their cloud infrastructures [1]–[4]. The use of FPGAs in cloud has dramatically improved their computing capabilities [1], [5], [6]. However, an FPGA instance can be underutilized when allocated to a single user, leading to FPGA resource wastage. Therefore, the cloud servers can deploy multi-tenanting to avoid this wastage. Multi-tenanting allocates each FPGA instance to more than one user leading to increased FPGA utilization and profit margin [7]. However, in hindsight, it introduces several security vulnerabilities in cloud—the attackers leverage these vulnerabilities to perform several attacks, namely, power side-channel (PSC) attack [8]–[13], fault attack [14]–[18], and covert channel attack [19]–[22].

Researchers have proposed several works demonstrating PSC attacks on cloud FPGAs. The works [9] and [11], demonstrates the CPA attack on a 128-bit AES design. Likewise, the work [8] demonstrates a simple power analysis (SPA) attack on RSA for a threat model considering cloud FPGAs. Similarly, there have been works proposing defenses to thwart PSC attacks on AES [9], [23], [24]. However, it is important to note that in order to perform any valid/accurate security assessment, it is imperative that we consider realistic scenarios of the underlying application and also have access to attack vectors that compromise the security guarantees of the underlying application. Hence, we focus on addressing the challenges in the existing attack and defense works. The following section discusses these challenges.

1.2 Challenges in PSC Attack on Cloud FPGAs

Here, we discuss the specific challenges that we encounter for PSC attacks on cloud FPGAs. The two primary challenges in the existing attack techniques are (i) repeatability of the attack, and (ii) the impact of sensor design and placements. These challenges are discussed next in detail.

Repeatability. To demonstrate the efficacy of any defense technique, we need an attack technique that is 100% repeatable, i.e., launching the attack every time should retrieve the entire key. If the attack is not repeatable, then the increase in MTD may be due to improper implementation of the attack technique and not due to the proposed defense technique. The previous works either provide only 42% attack repeatability rate [9] or do not study this

- *N. G. Jayasankaran is with Qualcomm India, KA, 560066. E-mail: gjn@qti.qualcomm.com*
- *H. Guo, S. Patnaik, and J. Rajendran are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, 77843. E-mail: {guohao2019, satwik.patnaik, jv.rajendran}@tamu.edu*
- *J. Hu is with the Department of Electrical and Computer Engineering and the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, 77843. E-mail: jianghu@tamu.edu*

aspect [11].

The impact of TDC-based sensor's placement and noise. The sensor's placements and the impact of noise on the efficacy of the CPA attack results must be studied, as these factors lead to an increase in the MTD. The attack feasibility reduces as the MTD increases. The increased MTD mandates the attacker to have access to an increased number of power traces. In prior works the defender analyzes the impact of the relative position of the TDC sensor with respect to the crypto core (AES). Additionally, the manual placements of TDC sensors that have been discussed in [24] and [25] focus on fine and coarse calibration settings. Here, the total path length of the TDC sensor can be made to have small or large variations, by fine and coarse tuning, respectively. However, in our work, as explained in Section 1.2 and Section 5, we analyze the impact of manually placing the FPGA primitives such as look-up tables (LUTs) and carry chain (CARRY4) that infers the TDC sensor. Further details regarding TDCs are discussed in Section 5.

1.3 Challenges in PSC Defenses on Cloud FPGAs

The challenges in the existing defenses are discussed next.

Realistic scenario. Crypto algorithms such as AES and RSA are a part of an SoC or wireless communication network. Nevertheless, there are no previous works that study the impact of additional circuits that reside along with the crypto cores.

Manual placement of FPGA primitives. The design implemented on the FPGA is inferred using FPGA primitives, such as look-up tables (LUTs), flip-flops, and carry chains. The impact of manually placing these primitives that infer the circuit-under-protection is not studied in the existing works.

Challenges in active fence defense. The existing cloud FPGA defense [23] uses ROs to reduce the SNR of the power traces collected by the TDC-based sensors. However, ROs fasten the FPGA aging by creating hotspots [14], [26] and also induce faults, making it vulnerable to fault injection attacks [10]. Additionally, cloud servers such as AWS blocks FPGA bitstream having combinatorial loops. Hence, implementing a defense using RO is not practically possible.

Defense thwarting only RO-based sensors. The current defenses [27] can only detect ROs implemented in the attacker's logic. However, a successful CPA attack on AES is achieved by using TDC-based sensors [9], [23], [24]. Hence, there is a need to secure the victim's logic from TDC-based sensors. Therefore, this work focuses on addressing the above challenges in the cloud FPGA domain.

1.4 Contributions and Organization of the Paper

The contributions corresponding to each of the challenges in the attack and defense domains are listed below.

- The impact of junction temperature on the MTD is studied. Maintaining this temperature within a limit aids in achieving a repeatable attack. This repeatability means launching the attack every time retrieves the correct 128-bit AES key.
- The sensitivity of the TDC-based sensors is improved by analyzing the impact of dissimilar net delays in the sensor design.
- The resilience to the CPA attack is evaluated in a realistic scenario. The defender's logic consists of AES, other crypto cores such as MD5 and SHA256, from the MIT-II common evaluation platform (CEP) [28] SoC platform.
- The impact of the placement of FPGA primitives inferring the crypto core is analyzed.

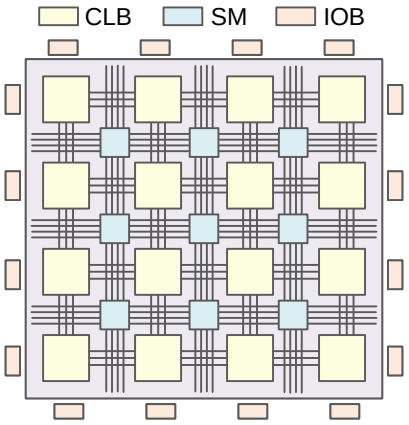


Fig. 1: Field-programmable gate array (FPGA) consists of configurable logic blocks (CLBs) connected to each other via switch matrix (SM). The FPGA communicates with the outside world using input-output blocks (IOBs).

- The realistic defense scenario considered in this work neither affects the reliability of the FPGA nor induces voltage faults.
- The defense scenario considered also thwarts CPA attack irrespective of the type of sensor used by the attacker.

The paper is organized as follows. In Section 2, we explain the background details related to PSC attack, sensors used, and the methodology of the CPA attack. In Section 3, we explain the previous work related to the PSC attack and its defenses in the cloud FPGA domain. The experimental setup used in this work is explained in Section 4. Section 5 discusses our enhanced PSC attack and the corresponding attack results. Section 6 shares the different methodologies proposed by this work to secure cloud FPGAs against PSC attacks and their experimental results. Finally, in Section 7, the inferences and conclusion from our experiments are explained.

2 Background

This section discusses FPGA fundementals, PSC attack, different sensors inferred using FPGA primitives, and CPA attack methodology.

2.1 Understanding FPGAs

As this work uses Xilinx Zynq ZC706 FPGAs, we discuss the FPGA fundamentals using Xilinx terminologies. The FPGA consists of a two-dimensional array of configurable logic blocks. The communication between these logic blocks is via switch matrices, as illustrated in Fig. 1. These switch matrices consist of programmable interconnects that route signals. Based on the FPGA device used, each configurable logic block consists of a fixed number of slices. The FPGA used in this work has two slices per logic block. Each slice consists of the FPGA primitives such as LUTs, multiplexers, carry logics (CARRY4), and flip-flops. In this FPGA, each slice is composed of eight LUTs, three multiplexers, one CARRY4, and eight flip-flops. The LUTs, multiplexers, and CARRY4 implement combinational logic, whereas the flip-flops implement sequential logic.

2.2 PSC Attacks on Cloud FPGA

In a conventional PSC attack, the attacker has physical access to the FPGA boards, where cryptographic algorithms, such as AES, are implemented. For a given key and a plaintext, the AES generates a corresponding ciphertext. Using an oscilloscope

connected to the power rail of the FPGA, the attacker measures the power consumed (power traces) during the execution of each step in the AES algorithm. First, the power traces are logged for different plaintexts and a fixed key. Following this, he/she runs the PSC attack on the collected power traces to determine the secret key [29]. Unlike the conventional setup, where the attacker has physical access to the FPGA board, he/she does not have this access in the cloud. Thus, in cloud FPGAs, the bitstream loading and debugging are done remotely. However, even without this access, researchers have successfully demonstrated PSC attack on the cloud FPGAs [8]–[12]. The following paragraph explains how the PSC attack is feasible even without physical access to the FPGAs on a cloud server.

A successful PSC attack requires the sensing of voltage drop in the power distribution network (PDN) shared between the attacker and the victim logic. The PDN connects the supply voltage to the FPGA primitives, such as LUTs and flip-flops. Irrespective of the change in load current to these primitives, the supplied voltage must be constant. However, due to the PDN structure's asymmetry, the supply voltage drops in the presence of load current variations [8], [9], [11], [24]. This variation in the load current is due to the data-dependent operations executed by the cryptographic algorithms. That is, the voltage drop on the PDN reflects the power consumed by these algorithms. The work [8] utilizes this relationship to successfully perform the simple power analysis attack on the 1024-bit Rivest, Shamir, Adleman (RSA) algorithm implemented on a cloud FPGA. Thus, even in the absence of physical access, the attacker uses a ring oscillator (RO) or a time-to-digital converter (TDC) circuits constructed using LUTs and flip-flops to measure the voltage drop rather than using an oscilloscope. The following section details the different sensors used in previous works launching PSC attack on cloud FPGA.

2.3 Sensors Used in Cloud FPGAs

The most commonly used sensors for performing the PSC attack on cloud FPGAs are ROs [8], [12] and TDCs [9], [11], [24]. These sensors sense the voltage drop during the execution of the cryptographic algorithms. This sensing is feasible, as the voltage supplied to the logic gates has an inverse proportionality impact on the propagation delay of these gates [30]. Therefore, a change in the propagation delay reflects the voltage drop. We now explain the different sensors using which the attackers measure the voltage drops in the FPGAs.

2.3.1 Ring Oscillators (ROs)

The ROs are widely implemented in cloud FPGAs to aid PSC attacks [8], [12], thermal covert-channel attacks [26], voltage covert-channel attacks [19], [22], [25], [31], and attacks based on inducing timing constraints violations. The conventional ROs are inferred using the LUTs. Hence, when the power consumption around these LUTs changes, the voltage drop varies. This variation gets reflected as a change in the propagation delay in these LUTs [30]. Therefore, the change in this propagation delay affects the time period and hence, the frequency of oscillations of the ROs. These ROs introduce several vulnerabilities listed below in the cloud FPGAs:

- In [26], ROs are used as heat generators to aid the formation of thermal covert channels. As multiple users share a single FPGA, this work shows that the heat generated by one user can be observed by another using the same FPGA instance at a different time.
- The ROs generate severe voltage fluctuations in [14]. These fluctuations crash the FPGA within a few microseconds. As a result, the system can be used only after power-cycling.
- They support voltage covert channel implementation [25].
- the attacker can induce timing constraint violations in the defender logic using ROs [17]. These faults are critical as they are temporary and hence, impossible to detect.

As ROs introduce various vulnerabilities in cloud FPGAs, the cloud service providers such as AWS blocks FPGA bitstreams containing combinatorial loops from getting deployed [33]. However, researchers designed ROs using latches and flip-flops to escape this filter in the Amazon firewall [27], as illustrated in Fig. 2(b) and 2(c).

2.3.2 Time-to-Digital Converters (TDC)

TDCs are also used in PSC attacks [9], [11], [24] to sense voltage drop variations. These sensors were originally developed to measure the propagation delay of logic gates (ASIC) and LUTs (FPGA) [34]–[37]. As the change in propagation delay reflects the change in voltage drop [30], the TDC design is leveraged to sense voltage variations. They can measure the time interval between two signal pulses or the arrival time of a single pulse and then convert it to a digital number. They are used in [27], [32] to measure voltage fluctuation in FPGAs. In [25], the TDC finds its application in the receiver side to sense the voltage change. Also, in [17], it is used as a voltage drop sensor. In this work, TDC-based sensors measure the dynamic power consumed by the 128-bit AES design.

We supply a clock signal to the chain of CARRY4 primitives, as shown in Fig. 3. Each CARRY4 primitive consists of four buffers connected in series, and the voltage drop experienced by these buffers influences the propagation delay experienced by them. This delay, in turn, controls the number of buffers the clock signal travels through the chain. Finally, a latch controlled by the same clock signal gets connected to each buffer's output to log the number of buffers the clock has traveled. As TDCs can measure

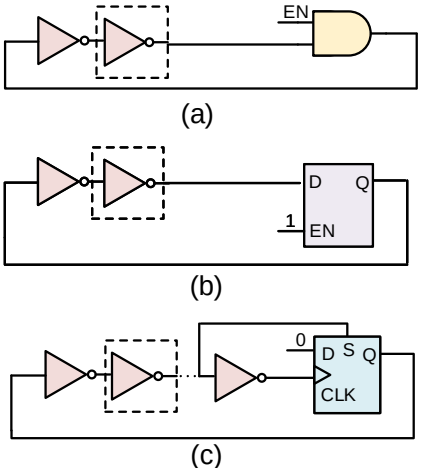


Fig. 2: Ring oscillator (RO) variants used in the prior works [8], [12], [14], [17], [19], [25]–[27], [31], [32]. The dotted box corresponds to an even number of inverters. (a) A simple RO based on a combinatorial loop. A single LUT infers each of these inverters in the FPGA. (b) RO based on the latch, which avoids the formation of a combinatorial loop. (c) Another RO variant, based on flipflops that avoids the formation of combinatorial loops. This variant can escape Amazon AWS firewall filters that block FPGA bitstreams with combinatorial loops.

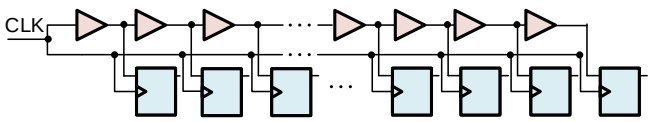


Fig. 3: Time-to-digital converter.

nanosecond delays and has higher accuracy [38], they are used in measuring the power consumed by the AES design. Hence, in this work, we deploy TDCs to perform the same functionality. Apart from the impact of dynamic power consumption on the TDC sensor's output, this output is also affected by process variations and temperature. We have not studied the impact of process variation and temperature on the TDC output in the current work and will explore these impacts in our future works. The following section explains the CPA attack methodology.

2.4 CPA Attack Methodology

This work uses the Hamming distance model based CPA attack, proposed in [39]. A successful attack on a 128-bit AES design involves the following steps:

1) Using the TDC-based sensors, the attacker measures the power consumed by the AES for encrypting different plaintexts. The number of power traces required by the attacker to determine the 128-bit key determines the strength of the attack — the lesser the number of power traces stronger the attack is.
2) For the first byte of the ciphertext, the attacker determines the output of the ninth round (first byte of intermediate ciphertext) using one out of 256 possible key guesses. He/She then calculates the Hamming distance between the first bytes of ciphertext and the intermediate ciphertext. Finally, the attacker repeats the Hamming distance calculation for the remaining 255 key guesses.
3) The attacker then correlates each of the power traces measured in step 1 with the 256 different values of Hamming distances calculated from the previous step.
4) The key guess corresponding to Hamming distance that has the maximum correlation with the power trace is the correct key byte.
5) The first byte of the ninth round's intermediate ciphertext and the key controls the first byte of ciphertext, i.e., the encryption of each byte of the intermediate ciphertext is independent of the other bytes. Hence, the attacker retrieves one key byte at a time. Therefore, the attacker repeats steps 2 to 4 for all the remaining 15 key bytes to determine the 128-bit AES key.

3 Previous Works

This section discusses the challenges in the previous works that proposed the PSC attack on 128-bit AES on a cloud FPGA platform and the defenses to thwart this attack.

3.1 PSC Attack on AES Implemented on Cloud FPGAs

Several works demonstrated the PSC attack on the AES implemented on the victim's side [9], [11]. In [9], a TDC-based sensor is used to measure the power consumed by the AES for each encryption. This work can retrieve the 128-bit AES key using $500k$ power traces. However, as mentioned in this work, the attack success rate is only 42%, i.e., the attack is not repeatable. Apart from demonstrating the PSC attack, another similar work [11] shows the impact of the distance between the attacker and victim on the number of traces required to retrieving the AES key. This work requires $1.8k$ traces to retrieve one key byte.

3.2 Defenses to Thwart PSC Attacks

Active fences [23]. This work proposes to reduce the signal-to-noise ratio (SNR) by increasing the noise in the AES encryptions. This reduction in SNR is achieved by enabling and disabling a group of ring oscillators (i) randomly and (ii) using the output value of the TDC-based sensor that senses the power consumption of the AES crypto under protection. With this defense, the MTD to determine one key byte is as high as $120k$ traces using the former method and $300k$ traces for the latter. In this technique, the ROs surrounds the crypto algorithms to form a fence. This implementation secures these algorithms against PSA. As explained in the work, the RO is implemented using a single look-up table (LUT) that infers a two-input NAND gate. The output of the NAND gate is fed back to one of the inputs forming the RO. The other input connects to the enable signal, which enables or disables the RO.

CPAmap [24]. This work aims to understand the underlying mechanisms and dependencies of chip-internal side-channel attacks. It investigates the sensitivity of the TDC-based sensors on different locations on the FPGA exhaustively. It is achieved by running the correlational power analysis (CPA) attack on the traces collected by the sensors placed at different locations. The impact of Vivado implementation settings on the bitstream generations is also studied. However, there are few challenges in this work, as mentioned below.

- After executing the PSC attack at multiple locations on the FPGA, the cloud service provider identifies the locations on the FPGA that are not safe for the crypto algorithms. These are the locations where the sensor can determine the key with less MTD. The provider refrains from allocating these locations to any user leading to FPGA resource wastage.
- The results of this work prove that unless an exhaustive experiment is conducted on each FPGA, the cloud service provider cannot determine the sensitive locations on the FPGA. This process is time-consuming, and it might have to be repeated as the chip ages, as aging affects the sensitivity of the FPGAs.

Mitigating voltage attacks [40]. The work proposes a defense against fault attacks. It senses or identifies the source of voltage attacks (logics that cause abnormal voltage drops) on the cloud FPGAs during runtime, i.e., when the attack is executed. The attacker implements malicious ROs triggers sudden voltage drops causing fault attacks [14]. To defend against this attack, the cloud service provider in [40] deploys sensors at multiple locations on the FPGA. If the voltage drop measured is less than the pre-defined value, the clock to that region is disabled, thereby thwarting the attacks. Though this defense is to secure against fault attacks, as it deducts the presence of ROs it can be leveraged to secure against PSC attacks due to ROs [8].

Mitigating electrical-level attacks [27]. This work is an offline countermeasure that translates the FPGA bitstreams to flat technology mapped netlist. This netlist is then parsed for the presence of circuits such as ROs and TDCs. If these circuits are detected in the netlist, the bitstream is blocked from being deployed on the cloud servers, as they aid PSC attacks.

4 Evaluation Plan and Setup

This section discusses the evaluation plan of this work. This work focuses on (i) improving the PSC attack and (ii) studying

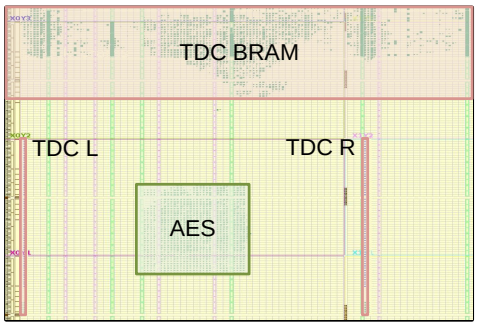


Fig. 4: FPGA device view of the power side-channel attack (PSA) setup. The victim's logic (green) is the 128-bit iterative AES crypto algorithm. The attacker uses two TDC-based sensors to measure the power consumption and the block RAMs (BRAMs) to store these power traces. The attacker's logic is shown in red.

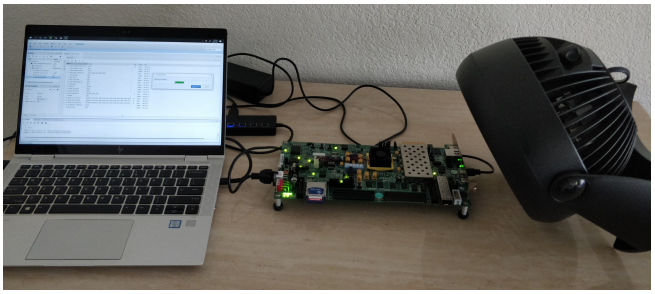


Fig. 5: All experiments in this work are demonstrated on the Zynq ZC706 FPGA evaluation platform. The board is connected to the host machine, where the power traces are collected to perform correlation power analysis (CPA) attacks. Apart from the fan on the heat sink to reduce the FPGA junction temperature, we also placed a table fan to cool the FPGA board further.

the impacts of primitive-level placement and extra logic placed along with the crypto design. Rather than proposing a new attack or defense technique, this work studies the impact of primitive-level placement in the sensor design (attack) and the crypto design under protection (defense). Also, we evaluate the CPA attack on multiple case studies where other logic designs such as processors and filters are placed along with the AES under protection. Our baseline experimental setup consists of two TDC-based sensors, one on the left and one on the right of the AES, as illustrated in Fig. 4.

4.1 Threat Model

This work assumes that the cloud services support multi-tenanting, i.e., multiple users can share one FPGA instance, in which one of the users can be an attacker (a malicious user). As the cloud services correspond to a common resource pool shared by multiple users, there is a high probability that the allocation of the victim's logic and the attacker's (malicious user) logic happens in the same FPGA instance. Additionally, the attacker or defender does not have physical access to the FPGAs. Therefore, he/she cannot probe the power rails to measure the power consumed using oscilloscopes.

This work chooses the 128-bit iterative version of the AES crypto algorithm as the victim's logic, whose source code is available at [41]. It also includes a virtual input output (VIO) debug IP core [42] to transmit and receive data from and to the AES core via the JTAG signals of the FPGA. The attacker's logic uses the TDC-based sensors to measure the power consumed by the AES encryption. Additionally, the attacker's logic includes block RAMs and a VIO debug IP core to store and send the power traces, respectively, to the user. The source code of the CPA attack [43] is tailored to suit our attack setup. This work uses the Zynq ZC706 evaluation platform to execute all our experiments. This platform is unmodified and we have not removed or added any capacitors on the evaluation platform. As our threat model focuses only on remote PSC attacks, where the attacker does not have access to the physical pins of the FPGA or the capacitors associated with the power supply, the evaluation platform is used without any modifications for conducting all our experiments. This evaluation platform along with the host machine and the cooling fan is shown in Fig. 5. The following section shares our attack contributions and the CPA attack results on our enhanced attack setup. Similarly, the CPA attack results, power, and area consumption of the different placement-based experiments and experiments with extra logic follow this section.

5 Our Enhanced Attack on 128-bit AES

This section focuses on setting up a repeatable attack to evaluate the resilience offered by our defense securing cloud FPGAs against PSC attack, i.e., the crypto key must be retrieved every time launching the attack on the cloud FPGAs. If the attack is not repeatable, then the increased minimum traces to disclosure (MTD) may be due to improper implementation of the attack itself and not the proposed defense. However, the previous works [9], [23], [24] either do not have a 100% attack success rate or do not evaluate the success rate. Hence, in this work, we propose fine-tuning the sensor design to reduce the MTD required to determine the 128-bit AES key and make the attack repeatable. The following sections discuss (i) the manual placement of FPGA primitives inferring the sensors, (ii) determining the time instant to which the CPA attack has the highest correlation, and (iii) studying the impact of junction temperature.

Sensor Placement. In this work, we explore the impact of manually placing each FPGA primitive inferring the TDC-based sensor. As explained in Section 2, the TDC consists of a chain of CARRY4 primitives. Each primitive consists of four outputs, where each output gets connected to a latch. These latches are, in turn, connected to the flip-flops. When Vivado places these primitives automatically, the net delay between the latches and their corresponding flip-flops are different for different pairs, as shown in Fig. 6 (b). This difference impacts the sensor output. By manually placing the flip-flops (colored in blue) illustrated in Fig. 6 (c), the net delay is approximately equal between all the latch–flip-flop pairs. This manual placement of these latches and flip-flops is achieved by providing user-defined constraints, such as LOC and BEL, during bitstream generation [44]. With the help of this primitive-level placement, we could retrieve the 128-bit AES key using $10.7k$ and $10k$ traces using the left and right TDC-based sensors, respectively.

Determining the Time Instant at which the Highest Correlation Occurs. As explained in Section 2, in the CPA attack, the key guess corresponding to Hamming distance that has a maximum correlation with the power trace is the correct key byte. Here, the Hamming distance is calculated between the tenth and ninth round ciphertexts. Hence, rather than considering the power consumed during all the ten rounds of AES encryption, we consider only the power consumed by the flip-flops during the tenth round of encryption. As a result, it helps achieve a low MTD ($2.3k$ and $2.1k$ traces using the left and right TDC-

Fig. 6: (a) FPGA device view of the TDC-based sensor. The zoomed version of the section of sensor enclosed in the red box are shown in Fig. 6 (b) and (c). (b) The Vivado tool automatically places the FPGA primitives corresponding to the sensor, leading to unequal net delay between the different latch-flipflop pairs. (c) The attacker judiciously places the FPGA primitives such that there is an approximately equal net delay between the different latch–flip-flop pairs.

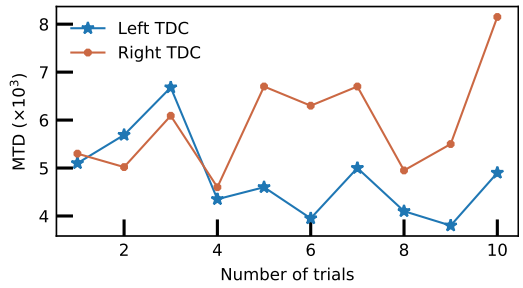


Fig. 7: Correlation power analysis (CPA) attack results on the enhanced attack setup with two sensors, left and right TDCs. The reliability of the attack is tested by collecting the traces multiple times for the same build. In each trial, all 16 key bytes are retrieved successfully. Minimum number of traces for disclosure (MTD) and time to digital converter (TDC).

based sensors, respectively) compared to the setup, where the power traces corresponding to all the ten rounds are considered ($10.7k$ and $10k$ traces using the left and right TDC-based sensors, respectively). Additionally, it also reduces the number of samples collected for each encryption. We have added this section to understand what MTD the attacker can achieve in a best-case scenario where he/she uses only the power traces corresponding to the ninth and tenth rounds. Our results show that the MTD is lesser when only these rounds are considered rather than all 10 rounds of power traces, i.e, the power traces corresponding to the other rounds act as noise and hence increase the MTD to determine the 128-bit key.

In this work, by fine-tuning the sensor design, we propose to demonstrate a repeatable attack on the 128-bit AES algorithm implemented on the Zynq ZC706 evaluation board. As shown in

TABLE 1: Correlation power analysis (CPA) attack results on our enhanced design. The design consists of AES (victim's logic) and the time-to-digital converter (TDC) (attacker's logic). The TDC is used for sensing the voltage variations in the victim's logic. The impact on temperature is studied. Minimum number of traces for disclosure (MTD).

Build number	No cooling time		30 minutes cooling time	
	bytes recovered	MTD	bytes recovered	MTD
1	16	5400	16	3900
2	15	8471	16	4600
3	15	5977	16	5100

Fig. 7, the 128-bit key is retrieved using $3.8k$ traces. Furthermore, on all the ten trials, the attack successfully determines the 128-bit key. As explained in Section 5, the attack setup consists of the iterative AES algorithm and two TDC-based sensors, one on each side of AES. This setup is illustrated in Fig. 4. The MTD reported by the previous works to determine the AES key is either around $1.8k$ traces [23] for one key byte or $500k$ for 16 key bytes. However, as illustrated in Fig. 7, the minimum and maximum MTD to determine the 128-bit key are $3.9k$ and $8.1k$ traces, respectively. The manual placement of each primitive in the sensor design aids in reaching this low MTD. This placement ensures approximately equal net delays between each latch–flip-flop pair, thereby reducing the errors in the sensor output, reflecting the power consumed by the defender's logic.

Understanding the Impact of Junction Temperature. Powering on the evaluation board for a long time increases the junction temperature of the FPGA. Additionally, the rate at which the different parts of the FPGA cool depends on these heat sink paths [45]. Hence, the increased junction temperature induces temperature-dependent noise. This noise impacts the CPA attack resulting in increased MTD to determine the 128-bit key. As shown in Table 1, with increased junction temperature, the CPA attack requires a higher MTD to retrieve all key bytes. However, maintaining this temperature below 30°C (using a cooling fan), the CPA attack retrieves all key bytes within $6k$ traces. The increase in junction temperature is already mitigated by placing the evaluation board in the refrigerator in [24]. However, our objective to study the impact of junction temperture on the MTD is to show that we could achieve a repeatable attack.

Table 2 lists the existing works on PSC attack along with this work. The attack implemented in this work has 100% repeatability. Additionally, by manually placing the sensor primitives we could achieve a very low MTD of 3800 traces compared to the sensor design that was automatically placed by the Vivado software that requires an MTD of $27k$ traces.

6 Our Defense Analysis

6.1 The Impact of Primitive-Level Placement of AES on CPA Attack

As given in Section 2.4, for the correct key guess, the correlation between the power trace and the Hamming distance is maximum. This Hamming distance corresponds to the number of bit-flips between the AES's ninth and tenth (ciphertext) round outputs. These bit-flips are the change in flip-flop outputs that are responsible for the power consumption. Unlike the LUTs, the flip-flops (edge-sensitive) output changes at the rising edge of the clock. Thus, the flip-flops in the AES predominantly contribute to dynamic power consumption compared to the LUTs. As flip-

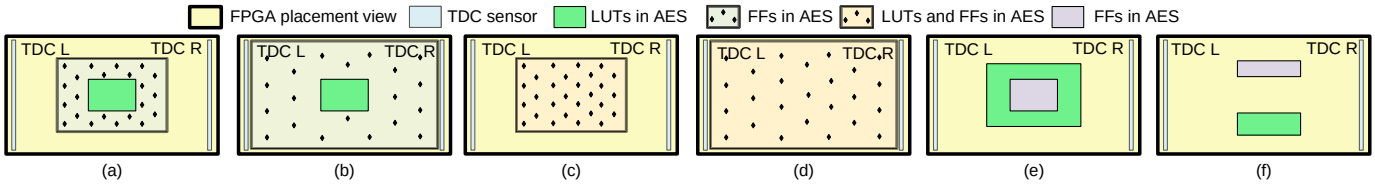


Fig. 8: Placement strategies. (a) The flip-flops in the AES design have three empty slices between each other along the X and Y axes. (b) The flip-flops in the AES design have six empty slices between each other along the X and Y axes. (c) The flip-flops and LUTs in the AES design have three and two empty slices, respectively between each other along the X and Y axes. (d) The flip-flops and LUTs in the AES design have six and four empty slices, respectively between each other along the X and Y axes. (e) The LUTs surround the flip-flops in the AES design. (f) The flip-flop block (violet) and the LUT block (green) are placed one above the other.

TABLE 2: The PSC attack results on different FPGA platforms reported by the works [9], [11] and this work. Minimum number of traces for disclosure (MTD). Power side-channel (PSC) attack. Not applicable (NA). Evaluation platform (EP).

Attack property	PSC attack [9]	Inside job [11]	This work
Repeatability	42%	NA	100%
MTD	$500k$	1800	3800
# of key bytes retrieved	16	Not available	16
Platform	Ultrascale+	XC6SLX75	ZC706 EP

flops mainly contribute to dynamic power consumption, we study the impact of the placement of these FPGA primitives manually on the CPA attack, rather than allowing the Vivado to perform automated placement of these primitives. Vivado places the FPGA primitives corresponding to the same logic as close as possible to reduce the wirelength delay. However, in this work, to ensure that the sensors cannot sense all the flip-flops of the AES, the defender spreads out the flip-flops across the clock region. The following are the different placement strategies the primitives in the AES algorithm have been placed on the Zynq 7000 series FPGA.

1) **Only the flipflops are spread out.** In this technique, only the flipflops in the AES design are spread across the clock regions, as illustrated in Fig. 8(a) and 8(b). The flops correspond to the plaintext and the key registers in the crypto algorithm. Increasing the number of empty slices between each flop along the X- and Y-axis of the FPGA achieves different spread levels.
2) **Both the flip-flops and LUTs are spread out.** In this technique, both the LUTs and flip-flops in the AES design are spread across the clock regions, as illustrated in Fig. 8(c) and 8(d). Increasing the number of empty slices between each flop and LUT along the X- and Y-axis of the FPGA achieves different spread levels.
3) **Flip-flop block surrounded by the LUT block.** Here, the flip-flops constrained within a block are surrounded by the LUTs, as shown in Fig. 8(e).
4) **Flip-flop block and the LUT block are placed one above the other.** Like the previous placement, the flip-flops and the LUTs are constraints to an individual block of regions on the FPGA and placed one above the other. Here, different spread levels are generated with an increasing number of empty slices between the two blocks, as illustrated in Fig. 8(f).

We now discuss the CPA attack results for these different placement strategies explained above.

6.2 CPA Attack on Our Defense Case Studies

6.2.1 *Impact of Primitive-Level Placement of AES on the MTD*

Spreading the flipflops and LUTs. The CPA results on the build based on the different primitive placement strategies discussed in Section 6.1 are illustrated in Fig. 9. When there are two empty slices between any two flip-flops, most of the key bytes are retrieved using less than $5k$ traces, as illustrated in Fig. 9(a). Increasing the empty slices to three requires around $10k$ traces to determine 16 and 10 key bytes by the right and left sensors, respectively, as shown in Fig. 9(b). However, increasing the number of empty slices between the AES flip-flops has different effects on the left and right sensors.

Fig. 9(c) shows that the left sensor determines the 128-bit key using less than $3k$ traces, while the right sensor retrieves less than three key bytes using $11k$ traces. Similarly, as shown in Fig. 9(d), when the number of empty slices between the LUTs and flip-flops of the AES is 2 and 3, respectively, the left sensor provides more accurate power traces compared to the right sensor. This difference is because the left sensor requires less than $2k$ traces to retrieve the 128-bit key, whereas the right sensor requires more traces to determine the 128-bit key. When the number of empty slices between the LUTs and flip-flops of the AES is increased to 4 and 6, respectively, the right sensor is more sensitive than the left sensor, as shown in Fig. 9(e). The above results show the impact of inhomogeneity in the PDN structure on the sensitivity of the TDC sensors.

Grouping flip-flops and LUTs in separate partitions. As discussed in Section 6.1, the flipflops and the LUTs of the AES design are grouped into separate partitions. In the AES design, 128 flip-flops and around $1.5k$ LUTs corresponding to the ciphertext. The partition block holding the flip-flops has a width of 30 slices and depth of four slices. Likewise, the block holding the LUTs has a width of 30 slices and a depth of 13 slices. There is a total of 75 empty slices between these partition blocks. As illustrated in Fig. 9(h), the left sensor's sensitivity is very high, as it can determine the 128-bit key with $1k$ traces. However, the right sensor could not retrieve more than two key bytes. Increasing the block width to 34 slices and reducing the block depth to three slices increases the sensitivity of both the sensors enabling the sensors to determine all the 16 key bytes, as shown in Fig. 9(l).

In the following section, we explain the impact of neighboring logic on the CPA attack.

6.3 Understanding the Impact of Neighboring Logic on PSC Attack

In this section, we evaluate the resilience against PSA in a practical scenario. Generally, the crypto algorithms such as

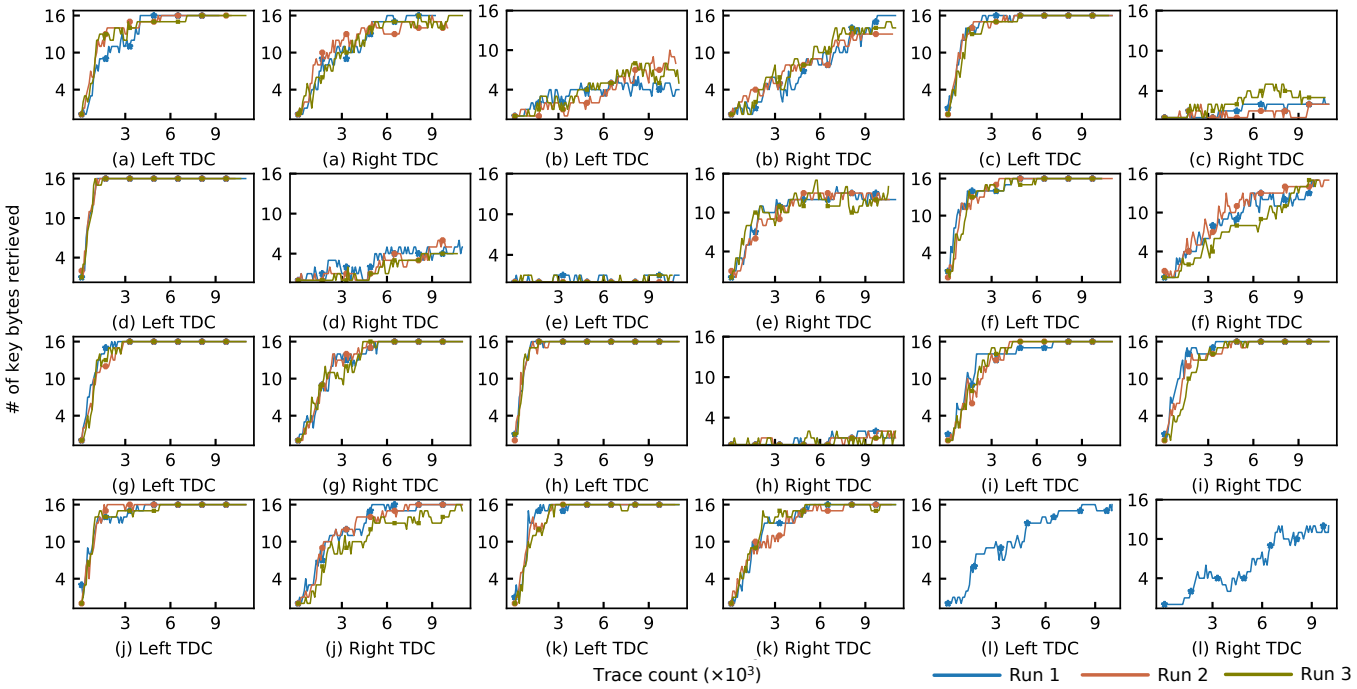


Fig. 9: Correlation power analysis (CPA) attack on the power traces collected by the left and right TDC-based sensors. The flipflops in the AES design have (a) two, (b) three, and (c) six empty slices between each other along the X and Y axes. The flipflops and LUTs in the AES design have (d) three and two empty slices, (e) six and four empty slices, respectively, between each other along the X and Y axes. (f) The LUTs surround the flipflops in the AES design. The placement of the flipflop block is above the LUT block. There are (g) 22, (h) 75, and (i) 113 empty slices in between the blocks. The width and depth of the flipflop block are 30 and 4, respectively. The width and depth of the LUT block are 30 and 13, respectively. There are (j) 75 and (k) 113 empty slices in between the blocks. The width and depth of the flipflop block are 90 and 2, respectively. The width and depth of the LUT block are 90 and 6, respectively.

AES reside along with other designs on the FPGA rather than standalone.

Choice of Logic Residing Along with AES. This crypto algorithm is used in different applications such as wireless security, processor security, file encryption, and SSL/TLS [46]. Therefore, in this work, we choose the following designs to reside along with AES: (i) the Kalman filter, (ii) the M32632 processor, and (iii) MIT-II CEP crypto cores. As filters are used predominately in wireless communications, they are chosen as one of the designs to study the resilience against PSA. An open-source processor design, M32632, is another logic chosen for this study. MIT-II CEP [28] is developed to provide an open-source evaluation platform to the users to evaluate their custom tools and techniques. As this platform has a processor integrated with crypto cores, we implement that section of the CEP with the crypto cores that include MD5 (Merkle–Damgård), SHA256 (Secure Hash Algorithm 256), DES3 (Data Encryption Standard), and RSA, along with the AES under protection.

Kalman Filters. Our first set of experiments were with Kalman filters. As discussed later in this section, the Kalman filter is either placed below or next to the AES. The FPGA device view of these placements are shown in Fig. 10(a), Fig. 10(b), and Fig. 10(c). The impact of spreading the flip-flops is studied in these experiments as well. In one build, the flip-flops in the Kalman filter are spread. While in the other, the flip-flops in the AES design are spread over more than one clock region.

Evaluation Platform. As the crypto algorithms such as AES are used predominantly in processor security, we will be evaluating the resilience of the build with an open-source processor M32632 and the MIT-II CEP [28]. In the case of MIT-II CEP, in this work, we implement only a part of this platform that includes all the crypto cores. We analyze the activity factor of each crypto core in the platform. This work assumes the best-case scenario for the attacker — he/she can place the sensors next to the boundary of the defender's logic. Also, the impact of the activity factor over the sensors' output is maximum when they are closest to the sensors [16]. Hence, in this work, the cores having high activity factors are placed close to the boundary of the allocated region, as shown in the device views in Fig. 11.

We shall now discuss the impact of placing these extra logic design along with AES on the MTD. Fig. 10 shows the device view of different experiments with Kalman filter along with the CPA attack results for each of these experiments.

1) Fig. 10(a) illustrates the device view with one Kalman filter with an input size of 16 bits. However, the CPA attack can retrieve 13 key bytes in less than $6k$ traces. A single Kalman filter is not sufficient to lower the SNR of the AES encryption. Thus, leaving it still vulnerable to PSA.
2) Fig. 10(b) One Kalman filter is placed between the AES and each of the sensors. Each Kalman filter has a 16-bit input. This setup requires $10k$ power traces to determine ten key bytes, approximately twice the number of traces compared to the previous setup with one Kalman filter.
3) Fig. 10(c) shows considerable increase in MTD. This design has only one Kalman filter; however, the input size is 48 bits.
4) Fig. 10(d) illustrates the CPA attack results when the M32632 processor and a 16-bit Kalman filter reside along with the AES.
5) From Section 6.2.1 it is evident that spreading the flip-flops of the AES reduces the MTD considerably up to $1k$ traces. As

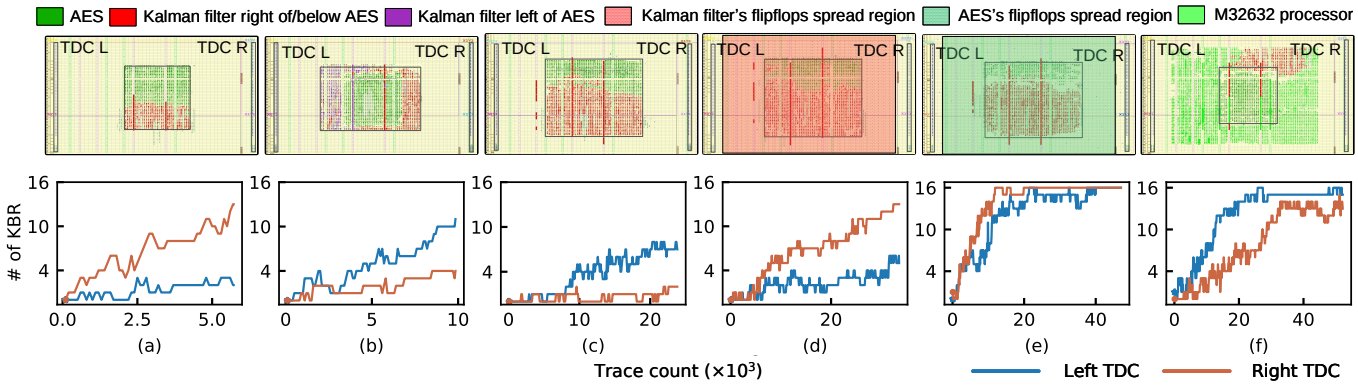


Fig. 10: The FPGA device view and the correlation power analysis (CPA) attack results on the different experiments with a Kalman filter and AES. (a) One Kalman filter with 16-bit input placed below the AES. (b) One Kalman filter with 16-bit input is placed on each side of AES. (c) One Kalman filter with 48-bit input placed below the AES. (d) This setup is similar to Fig. 10(c). Here, the Kalman filter's flipflops are manually spread over multiple clock regions. (e) This setup is similar to Fig. 10(c). Here, the AES's flipflops are manually spread over multiple clock regions. (f) This setup has an M32632 processor and Kalman filter along with AES.

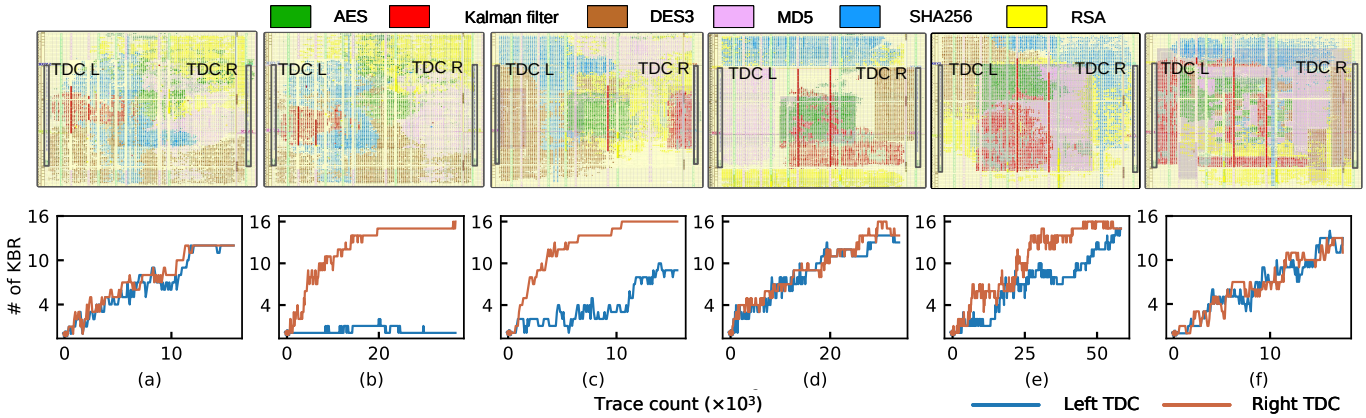


Fig. 11: The FPGA device view and the correlation power analysis (CPA) results on the different experiments with MIT-II CEP [28] residing along with AES. (a) The Vivado does the automatic placement of the MIT-II CEP cores. (b) The flipflops of the MIT-II CEP cores are placed in the same partition block in which AES resides. (c) The Kalman filter is placed between the right of AES and the right sensor. Similarly, DES3 is placed between the left of AES and the left sensor. (d) The activity factor of all the subblocks inside the MIT-II CEP is aligned with the AES activity factor. (e) As the Kalman filter has high activity factor compared to the other logic design in the MIT-II CEP cores, a 48-bit input Kalman filter is included in the design. (f) The defender manually does the placement, where the AES and MIT-II CEP cores are merged.

the wirelength between the flip-flops and LUTs increases, the power consumed also increases. Thus, in this setup, the flip-flops of the Kalman filter are spread out. This setup increases the power consumed by the filter. Thus, we intend to increase the power consumed by the Kalman filter in the total power consumed by the defender. However, the attack results are similar to those corresponding to the setup that does not have the Kalman filter flip-flops spread out.

6) As the wirelength between the flip-flops and LUTs increases, the power consumed also increases. In some cases, as the AES flip-flops are spread out, the sensor cannot sense the power consumption of a few of these flip-flops. The phenomenon will increase the MTD. Thus, the AES flip-flops are spread in the setup that has the Kalman filter with 48-bit input. In this setup, the SNR of the AES operation increases due to the increased power consumption from the additional wire length. Hence, as illustrated in Fig. 10(e), the CPA attack requires less than $11k$ traces to determine all 16 key bytes.

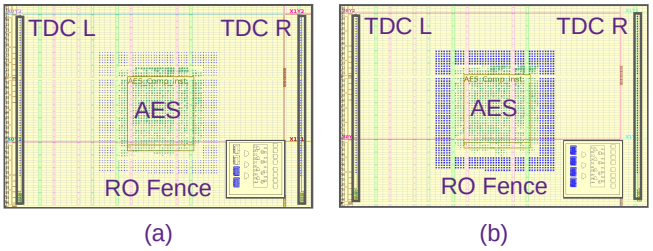


Fig. 12: FPGA Device view of the active fence implementation [23]. Each slice in the RO fence consists of (a) two and (b) eight ROs. Each device view has an inset figure showing the LUT utilization of the slice.

6.4 Active Fence Implementation [23]

To compare our technique with the existing works, in this section, we implement the active fences defense proposed in [23]. We then compare the CPA attack results of our work and this work. The AES implemented on Zynq 706 evaluation board requires 896 slices. Hence, as per the build details in [23], 896 ROs

are implemented around the AES. This implementation results in 12.5% LUT utilization per slice, as each slice accommodates eight LUTs. Apart from the design shared in the paper, we have tried other implementations of the ROs to check their impact on the CPA results. The other combinatorial RO design tried are as follows:

- As each slice consists of eight LUTs, two single-LUT ROs are implemented per slice, i.e., 1792 ROs implemented in the RO fence surrounding the crypto design.
- All eight LUTs implement the single-LUT RO. This build will degrade the FPGA as the fence has around $7k$ ROs, each oscillating at a frequency of around $1.4GHz$.
- One RO is inferred using eight LUTs. Each of the seven LUTs infers a buffer, and the remaining one LUT infers an invertor. Finally, the eight LUTs are connected in a daisy chain fashion to implement a RO.
- Similar to the previous design, this design has each of the seven LUTs to infer an invertor, and the remaining one LUT infers a buffer.

6.4.1 Impact of Active Fence on the MTD

The CPA attack results on the different implementations of the active fence [23] shared in Section 6.4 are explored in this section. Fig. 13(a) plots the CPA results for the setup with RO fence only on the right of AES. Fig. 12(a) illustrates the device view of this setup. The attack on the power traces from both the sensors shares similar results though the RO fence is only to the right of the AES. Placing the same number of ROs around the AES as in Fig.12(b) reduces the MTD (requires less than $12k$ traces) to determine the 128-bit key. This low MTD is due to the reduced noise on each side of the AES induced by the RO. There is only one RO inferred in each slice in the builds so far. Increasing the number of ROs to two and eight per slice increase the MTD required to determine the 128-bit key, as illustrated in Fig.13(d) and Fig.13(e). This increase in MTD is due to the increased activity factor as the number of ROs per slice increases.

6.4.2 Need for RO Design Using Flip-flops

Cloud servers such as AWS block this design from getting deployed on the F1 instance, as this technique implements combinatorial loops [33]. As ROs are used in PSC attacks [8], timing fault attacks [17], and also degrades the life of the FPGA [14], these servers thwart the loading of bitstream if they detect ROs in these bitstreams. Hence, we have also implemented the flip-flop-based ROs in the fences. The oscillating frequency of this type of RO is $284.01MHz$, which is less than the oscillating frequency of ROs inferred using a single LUT, equal to $1.4GHz$. Thus, the activity factor of the flipflop-based ROs is less compared to the single LUT ROs. This reduction, in turn, means less dynamic power and hence, less noise due to the ROs. Additionally, the ROs with very high oscillating frequency can have a destructive impact on the FPGA device. This impact is because the maximum clock frequency supported by the FPGA device is 800 to 900 MHz. Hence, the routing paths in the CLBs connecting the data ports of LUTs and flip-flops can support a frequency of up to only 400 to 450MHz. Therefore, a build is generated with ROs based on flip-flops, as shown in Fig. 2(c). The CPA results on the power traces collected from this build retrieve 15 out of 16 bytes using $10k$ traces, as illustrated in Fig.13(f).

Table 3 compares the defenses to thwart PSC attack on AES on cloud FPGAs with the defense proposed in this work. As shown in this table, compared to the existing defense works, this case study based work has the following advantages: (i) study the impact of the practical environment on the CPA results, (ii) study the impact of FPGA primitive-level placements on the CPA results, (iii) does not include ROs in their defense, (iv) this defense secures against TDC-based sensors, and (v) has a higher MTD and retrieves less key bytes compared to the existing defenses. However, unlike [24], we do not test the effectiveness of the defense at multiple locations. The onboard experimental results show that the CPA results are susceptible to the PDN design and routing apart from the placements. Hence, as part of future work, we intend to understand the PDN design and study the impact of routing on the CPA attack. Following these experiments, we will test the defense at multiple locations on the FPGA.

7 Inferences and Conclusion

Impact of Asymmetry in PDN Structure. Fig. 9 and Fig. 11 shows the change in MTD with change in the placement of the FPGA primitives corresponding to the 128-bit AES design. Though the functionality of the implemented design is same, even a small change in the placement locations causes a change in MTD.

- **Spreading the Flip-flops.** As shown in Fig. 9(a), 9(b), and 9(c) increasing the number of empty slices between the flip-flops increases the difficulty of retrieving the AES key by the right sensor. However, the left sensor can retrive the key using 6K traces when the number of empty slices are two (minimum spread) and six (maximum spread).
- **Spreading the Flip-flops and LUTs.** Increasing the number of empty slices between the flip-flops and LUTs decreases and increases the difficulty of retrieving the AES key by the right and left sensors, respectively, as illustrated in Fig. 9(d) and 9(e).
- **Spreading the Flip-flops and LUT Blocks.** The Fig. 9(h)

TABLE 3: Comparison of our contributions with the existing defenses: (i) active fences [23], (ii) CPAmap [24] and (iii) votlage attack mitigation [40]. CPAmap [24] requires as high as 10M traces to determine one key byte. However, this is using the active fences [23] that implements combinatorial loop-based ROs. As these ROs cannot be implemented on cloud servers, we have mentioned "not applicable" (NA) for minimum traces for disclosure (MTD) for the CPAmap [24].

Defense property	[23]	[24]	[40]	Our analysis
Practical environment	✗	✗	✗	✓
FPGA primitive-level placement of AES	✗	✗	✗	✓
Absence of ROs	✗	✗	✓	✓
Protection against TDC-based sensors	✓	✓	✗	✓
Tested at multiple locations	✓	✓	✗	✗
MTD	$10k$	NA	NA	$24k$
# of key bytes retrieved	14	NA	NA	8

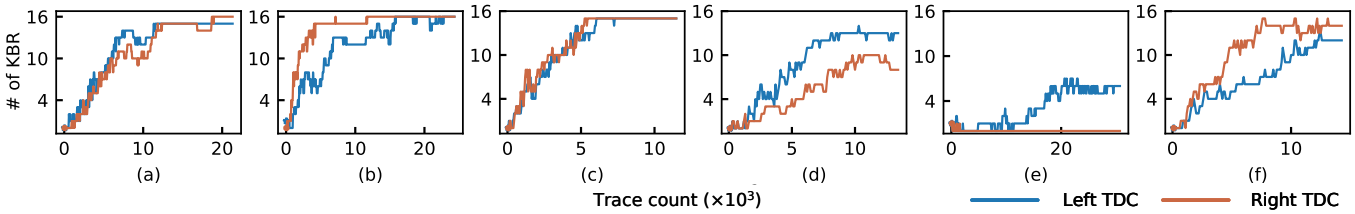


Fig. 13: Correlation power analysis (CPA) attack results based on the different flavors of active fences work [23]. A total of 896 slices implements the ring oscillators (RO). (a) A fence separates the AES and the TDC sensor made up of 896 ROs. Each slice has one ring oscillator (RO) implemented in a single look-up table (LUT). (b) The RO fence made up of 896 ROs surrounds the AES. The remaining builds have similar placement as in Fig. 12. The difference is in the number of ROs in each slice and RO design. (c) The RO is implemented using seven LUTs inferred as invertors, and one LUT inferred as a buffer. (d) Each slice has two ROs, and each of these ROs are implemented using one LUT. (e) Each slice has eight ROs, and each of these ROs are implemented using one LUT. (f) This setup has the RO design shared in Fig. 2(c).

corresponds to the CPA attack results for the build that has the partition block holding the flip-flops has a width of 30 slices and a depth of four slices. Likewise, the block holding the LUTs has a width of 30 slices and a depth of 13 slices. There is a total of 75 empty slices between these partition blocks. Here, we could not retrieve more than two key bytes from the traces collected from the right sensor. However, increasing the width of the flip-flop and LUT block by four slices in the build makes design vulnerable CPA attack, as shown in Fig. 9(l).

- **Placing the Flip-flops of the Surrounding Logic with AES.** The Fig. 11(b) corresponds to the CPA attack results for the build that has the flip-flops of MIT-II CEP crypto cores reside along with the FPGA. Here, the traces collected from the left sensor does not retreive even one key byte. However, using the traces collected by the right sensor we can retrieve all 16 key bytes using 40K traces.

Different Clock Regions. Fig. 4 shows that the left TDC and the AES share the same clock regions, X0Y0 and X0Y1; the right TDC is located in clock region X1Y0. Ideally, the left TDC must be more sensitive than the right one, determining the keys with lesser MTD. However, as shown in Fig. 11(b) and Fig. 11(c) the right TDC is more sensitive compared to the left one. Therefore, we cannot rely on the clock regions to understand the TDC sensitivities.

In this work, we address the different challenges in the PSC attack and its defenses. Firstly, we enhanced the attack by manually placing the FPGA primitives corresponding to the TDC-based sensor design. This manual placement helped achieve an MTD as low as 3.8K traces to retrieve the 128-bit AES key. The impact of the junction temperature on the MTD was studied, which helped set up a repeatable attack. As a result, our attack can determine the 128-bit key every time it is launched on the FPGA. We then studied the impact of spreading the FPGA primitives corresponding to the AES on the CPA attack results. Finally, we also evaluated builds with additional logic residing along with AES; compared their CPA results with builds supported with active fences. Our experimental results show that the AES with additional logic having sufficient activity factor can provide the same or increased MTD compared to the build with AES surrounded by the active fences. Additionally, our defense keeps the reliability of the FPGA intact as it does not include high-speed combinatorial loops.

REFERENCES

[1] Amazon, "Amazon EC2 F1 Instance," https://aws.amazon.com/ec2/instance-types/f1/, 2016, Last accessed on 09/05/2021.

[2] Microsoft, "Microsoft Azure," https://azure.microsoft.com/en-gb/, 2021, Last accessed on 09/05/2021.

[3] IBM, "CloudFPGA," https://www.zurich.ibm.com/cci/cloudFPGA/, 2020, Last accessed on 09/05/2021.

[4] U. of Texas at Austin, "Texas Advanced Computing Center," https://www.tacc.utexas.edu/, 2020, Last accessed on 09/05/2021.

[5] C. Jin, V. Gohil, R. Karri, and J. Rajendran, "Security of Cloud FPGAs: A Survey," *arXiv*, 2020.

[6] Avnet, "Where is FPGA in Cloud Computing Today?" https://www.avnet.com/wps/portal/apac/resources/article/where-is-fpga-in-cloud-computing-today/, 2019, Last accessed on 09/05/2021.

[7] Z. István, G. Alonso, and A. Singla, "Providing Multi-tenant Services with FPGAs: Case Study on a Key-Value Store," *IEEE International Conference on Field Programmable Logic and Applications*, pp. 119–1195, 2018.

[8] M. Zhao and G. E. Suh, "FPGA-Based Remote Power Side-Channel Attacks," *IEEE Symposium on Security and Privacy*, pp. 229–244, 2018.

[9] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are Cloud FPGAs Really Vulnerable to Power Analysis Attacks?" *IEEE/ACM Design, Automation & Test in Europe*, pp. 1007–1010, 2020.

[10] G. Provelengios, D. Holcomb, and R. Tessier, "Power Distribution Attacks in Multitenant FPGAs," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 28, no. 12, pp. 2685–2698, 2020.

[11] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "An Inside Job: Remote Power Analysis Attacks on FPGAs," *Design, Automation & Test in Europe*, 2021.

[12] C. Ramesh, S. B. Patil, S. N. Dhanuskodi, G. Provelengios, S. Pillement, D. Holcomb, and R. Tessier, "FPGA Side Channel Attacks without Physical Access," *Annual International Symposium on Field-Programmable Custom Computing Machines*, pp. 45–52, 2018.

[13] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "Remote Inter-Chip Power Analysis Side-Channel Attacks at Board-Level," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–7, 2018.

[14] D. R. E. Gnad, F. Oboril, and M. B. Tahoori, "Voltage Drop-based Fault Attacks on FPGAs Using Valid Bitstreams," *IEEE International Conference on Field Programmable Logic and Applications*, pp. 1–7, 2017.

[15] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "FPGAhammer: Remote Voltage Fault Attacks on Shared FPGAs, suitable for DFA on AES," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, pp. 44–68, 2018.

[16] G. Provelengios, D. Holcomb, and R. Tessier, "Characterizing Power Distribution Attacks in Multi-User FPGA Environments," *IEEE International Conference on Field Programmable Logic and Applications*, pp. 194–201, 2019.

[17] D. Mahmoud and M. Stojilović, "Timing Violation Induced Faults in Multi-Tenant FPGAs," *IEEE/ACM Design, Automation & Test in Europe*, pp. 1745–1750, 2019.

[18] T. Sugawara, K. Sakiyama, S. Nashimoto, D. Suzuki, and T. Nagatsuka, "Oscillator without a Combinatorial Loop and Its Threat to FPGA in Data Centre," *Electronics Letter*, vol. 55, no. 11, pp. 640–642, 2020.

[19] I. Giechaskiel, K. Eguro, and K. B. Rasmussen, "Leakier Wires: Exploiting FPGA Long Wires for Covert- and Side-Channel Attacks," *ACM*

Transactions on Reconfigurable Technology and Systems, vol. 12, no. 3, 2019.
[20] I. Giechaskiel, K. Rasmussen, and J. Szefer, "Reading Between the Dies: Cross-SLR Covert Channels on Multi-Tenant Cloud FPGAs," *IEEE International Conference on Computer Design*, pp. 1–10, 2019.
[21] ——, "CAPSULe: Cross-FPGA Covert-Channel Attacks through Power Supply Unit Leakage," *IEEE Symposium on Security and Privacy*, pp. 909–922, 2020.
[22] I. Giechaskiel, K. Rasmussen, and K. Eguro, "Leaky Wires: Information Leakage and Covert Communication Between FPGA Long Wires," *ACM Asia Conference on Computer and Communications Security*, pp. 18–27, 2018.
[23] J. Krautter, D. R. Gnad, F. Schellenberg, A. Moradi, and M. B. Tahoori, "Active Fences against Voltage-based Side Channels in Multi-Tenant FPGAs," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–8, 2019.
[24] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "CPAmap: On the Complexity of Secure FPGA Virtualization, Multi-Tenancy, and Physical Design," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 3, pp. 121–146, 2020.
[25] D. R. E. Gnad, C. D. K. Nguyen, S. H. Gillani, and M. B. Tahoori, "Voltage-based Covert Channels in Multi-Tenant FPGAs," Cryptology ePrint Archive, Report 2019/1394, 2019, https://eprint.iacr.org/2019/1394.
[26] S. Tian and J. Szefer, "Temporal Thermal Covert Channels in Cloud FPGAs," *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, p. 298–303, 2019.
[27] J. Krautter, D. R. E. Gnad, and M. B. Tahoori, "Mitigating Electrical-Level Attacks towards Secure Multi-Tenant FPGAs in the Cloud," *ACM Transactions on Reconfigurable Technology and System*, vol. 12, no. 3, 2019.
[28] Massachusetts Institute of Technology, "Common Evaluation Platform," https://github.com/mit-ll/CEP, 2021, Last accessed on 09/05/2021.
[29] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology*, pp. 388–397, 1999.
[30] S. Pant, "Design and Analysis of Power Distribution Networks in VLSI Circuits," *Ph.D. dissertation, The University of Michigan*, 2008.
[31] I. Giechaskiel, K. B. Rasmussen, and J. Szefer, "Measuring Long Wire Leakage with Ring Oscillators in Cloud FPGAs," *IEEE International Conference on Field Programmable Logic and Applications*, pp. 45–50, 2019.
[32] D. R. E. Gnad, S. Rapp, J. Krautter, and M. B. Tahoori, "Checking for Electrical Level Security Threats in Bitstreams for Multi-tenant FPGAs," *IEEE International Conference on Field-Programmable Technology*, pp. 286–289, 2018.
[33] Amazon, "Combinatorial Loops in F1 FPGA," https://forums.aws.amazon.com/thread.jspa?threadID=264137, 2017, Last accessed on 09/05/2021.
[34] K. M. Zick, M. Srivastav, W. Zhang, and M. C. French, "Sensing Nanosecond-Scale Voltage Attacks and Natural Transients in FPGAs," *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2013.
[35] K. Cui and X. Li, "A High-Linearity Vernier Time-to-Digital Converter on FPGAs with Improved Resolution Using Bidirectional-Operating Vernier Delay Lines," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2019.
[36] H. Xia, G. Cao, and N. Dong, "A 6.6 ps RMS Resolution Time-to-Digital Converter Using Interleaved Sampling Method in a 29 nm FPGA," *Review of Scientific Instruments*, vol. 90, no. 4, p. 044706, 2019.
[37] J. Wang, S. Liu, L. Zhao, X. Hu, and Q. An, "The 10-ps Multitime Measurements Averaging TDC Implemented in an FPGA," *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, pp. 2011–2018, 2011.
[38] S. Moini, X. Li, P. Stanwicks, G. Provelengios, W. Burleson, R. Tessier, and D. Holcomb, "Understanding and Comparing the Capabilities of On-Chip Voltage Sensors against Remote Power Attacks on FPGAs," *IEEE International Midwest Symposium on Circuits and Systems*, pp. 941–944, 2020.
[39] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," *Cryptographic Hardware and Embedded Systems*, pp. 16–29, 2004.
[40] G. Provelengios, D. Holcomb, and R. Tessier, "Mitigating Voltage Attacks in Multi-Tenant FPGAs," *ACM Transactions on Reconfigurable Technology and Systems*, 2021.
[41] Tohoku University, "Cryptographic Hardware Prject," http://www.aoki.ecei.tohoku.ac.jp/crypto/web/cores.html, 2007, Last accessed on 09/05/2021.
[42] Xilinx, "Virtual Input/Output v3.0," https://www.xilinx.com/support/documentation/ip_documentation/vio/v3_0/pg159-vio.pdf, 2018, Last accessed on 09/05/2021.
[43] NUEESS Lab Northeastern University, "Side Channel Analysis Library," https://bit.ly/2Y6k0ZD, 2013, Last accessed on 09/05/2021.
[44] Xilinx, "Vivado Design Suite Properties Reference Guide," https://bit.ly/2XeABu1, 2020, Last accessed on 09/09/2021.
[45] ——, "Power Methodology Guide," https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/ug786_PowerMethodology.pdf, 2013, Last accessed on 09/05/2021.
[46] R. Thomas, "Advanced Encryption Standard (AES): What It Is and How It Works," https://bit.ly/3zOeQzd, 2020, Last accessed on 09/05/2021.

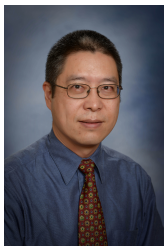
Nithyashankari G. Jayasankaran (S'18) received her B.E. degree in Electrical and Electronics Engineering from Anna University, Chennai, India, the M.S. degree in Microelectronics from Birla Institute of Science and Technology, Pilani, India, and her Ph.D. in Computer Engineering at Texas A&M University under the guidance of Prof. Jiang Hu and Prof. JV Rajendran. Her research interest is in the hardware security domain, emphasizing analog and mixed-signal circuits security and power side-channel attacks and defenses on cloud FPGA servers. Before joining Texas A&M, she worked on ASIC emulation, FPGA design, and board testing for base transceiver station based applications. Currently, she is working as an SoC security architect at Qualcomm, India.

Hao Guo (S'21) received her B. Eng in Electrical Engineering and Automation from Southwest Jiaotong University, Chengdu, China, in 2017 and the M.Sc. in Electrical Engineering from University of Southern California, Los Angeles, USA, in 2019. She joined the Texas A&M Secure and Trustworthy Hardware (SETH) Lab as a Ph.D. student in spring 2020. Her research lies in the hardware security domain focusing on FPGA security, logic locking, and applied machine learning for hardware security. She performed her summer internship at Cadence, Nanjing, China, in 2019 and worked at Qualcomm, Shanghai, China, in the R&D department as a DFT engineer before joining SETH Lab.

Satwik Patnaik received the B.E. degree in electronics and telecommunications from the University of Pune, India, the M.Tech. degree in computer science and engineering with a specialization in VLSI design from the Indian Institute of Information Technology and Management, Gwalior, India, and the Ph.D. degree in electrical engineering from Tandon School of Engineering, New York University, Brooklyn, NY, USA in September 2020.

He is currently a Postdoctoral researcher with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. His research delves into IP protection techniques, CAD frameworks for security, leveraging 3D paradigm for enhancing security, exploiting security properties of emerging devices, applied machine learning for hardware security, and side-channel evaluation.

Jeyavijayan (JV) Rajendran (S'09, M'15, SM'20) is an Assistant Professor in the Department of Electrical and Computer Engineering at the Texas A&M University. Previously, he was an Assistant Professor at UT Dallas between 2015 and 2017. He obtained his Ph.D. degree from New York University in August 2015. His research interests include hardware security and computer security. His research has won the NSF CAREER Award in 2017, the ACM SIGDA Outstanding Young Faculty Award in 2019, the Intel Academic Leadership Award, and the ACM SIGDA Outstanding Ph.D. Dissertation Award in 2017. He organizes the annual Embedded Security Challenge, a red-team/blue-team hardware security competition and has co-founded Hack@DAC, a student security competition co-located with DAC, and FOSTER. He is a member of IEEE and ACM.



Jiang Hu (F'16) received the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2001. He has worked with IBM Microelectronics, Armonk, NY, USA, from 2001 to 2002, and has been a Faculty Member with Texas A&M University, College Station, TX, USA. Dr. Hu received Best Paper Awards at ACM/IEEE Design Automation Conference 2001, IEEE/ACM International Conference on Computer-Aided Design 2011, IEEE International Conference on Vehicular Electronics and Safety 2018, and IEEE/ACM International Symposium on Microarchitecture 2021. He also received the IBM Invention Achievement Award and the Humboldt Research Fellowship. He has served on the editorial boards of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and the ACM Transactions on Design Automation of Electronic Systems. He was the Technical Program Chair of the ACM International Symposium on Physical Design 2011.