IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 5, 2021, accepted July 12, 2021, date of publication July 27, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100618

UCloD: Small Clock Delays to Mitigate Remote Power Analysis Attacks

DARSHANA JAYASINGHE, ALEKSANDAR IGNJATOVIC, AND SRI PARAMESWARAN
School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia

Corresponding author: Darshana Jayasinghe (darshanaj@cse.unsw.edu.au)

This work was supported by Australian Research Council (ARC) under Grant DP180104188.

ABSTRACT This paper presents UCloD, a novel random clock delay-based robust and scalable countermeasure against recently discovered remote power analysis (RPA) attacks. UCloD deploys very small clock delays (in the picosecond range) generated using the tapped delays lines (TDLs) to mitigate RPA attacks. UCloD provides the most robust countermeasures demonstrated thus far against RPA attacks. RPA attacks use delay sensors, such as Time to Digital Converters (TDC) or Ring Oscillators (ROs) to measure voltage fluctuations occurring in power delivery networks (PDNs) of Field Programmable Gate Arrays (FPGAs). These voltage fluctuations reveal secret information, such as secret keys of cryptographic circuits. The only countermeasure proposed thus far activates ROs to consume significant power and has managed to secure Advanced Encryption Standard (AES) circuits for up to 300,000 encryptions. Using TDLs available in FPGAs, UCloD randomly varies the clock to the cryptographic circuits under attack to induce noise in the adversary's delay sensor(s). We demonstrate correlation power analysis (referred to as CPA) attack resistance of UCloD AES implementations for up to one million encryptions. Compared to an unprotected AES circuit, UCloD implementations have minimal overheads (0.2% Slice LUT overhead and 4.8% Slice register overhead for Xilinx implementations and 0.5% LogicCells overhead for Lattice Semiconductor implementations).

INDEX TERMS Cryptography, encryption, security, side-channel attacks.

I. INTRODUCTION

Side channel analysis attacks [1] use emanated by-products of an encryption execution, such as power dissipation [2] (referred to as PA attacks), electromagnetic (EM) radiation [3], elapsed time [4] and cache hit/miss information [5], [6] to deduce the secret key from cryptographic algorithms. Side channel analysis attacks have shown successful at revealing secret keys from block cipher algorithms, such as AES [7] and Elliptic-curve cryptography [8] running on Application Specific Integrated Circuits (ASICs) [9], embedded processors [10], Graphics Processing Units (GPUs) [11] and Field Programmable Gate Arrays–FPGAs [12].

PA attacks on FPGAs employ an oscilloscope or an Analog to Digital Converter (ADC) to quantize the power dissipation (voltage or current fluctuations) emanating from the FPGA [13]. The newly proposed, Remote Power Analysis (referred to as RPA) attacks [14] uses a custom hardware design (referred to as a Delay Sensor) which acts as a sensor to detect voltage fluctuations occurring in the power delivery network (PDN) of FPGAs [15]. Voltage fluctuations occur in the PDN due to the power dissipation of other hardware designs, such as the AES circuitry executed along with the delay sensor on the FPGA. Due to the capacitance in the PDN, sudden power fluctuations can be detected at nanosecond or even picosecond time scales. The delay sensor readings will be affected due to such voltage fluctuations. In order to detect these voltage fluctuations, Time to Digital Converters (TDC) [14] and Ring Oscillators (ROs) [16] have been proposed. According to the authors in [16], multiple ROs have to be placed in the FPGA, while a single TDC sensor is able to reveal the secret key of AES [14]. Thus, we limit our focus to TDC sensors in this paper, since they are smaller and more efficient [17]. TDC sensor based RPA attacks have shown successful by placing the TDC sensor in most of the places of the FPGA floor plan [18]. RPA attacks have shown to be effective on the same FPGA where multiple users share the FPGA (known as multi-tenant FPGA architectures) [19], [20]. Such multi-tenanted FPGAs are expected to be common on cloud computers in future to reduce the cost

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

of cloud-based FPGA services [21]. Therefore, cloud FPGA services, such as Amazon EC2 FPGA cloud [22] and Alibaba FPGA cloud service, where FPGA acceleration is offered as a service, are vulnerable to RPAs [23]. It is imperative that countermeasures be deployed to prevent RPA attack vulnerabilities.

So far the only countermeasure proposed to prevent RPA attacks is the 'Active Fences' [24] which uses a set of ROs to dissipate random power consumption to conceal the voltage fluctuations which occur in the PDN. According to the authors in [24] after 300,000 encryptions, the secret key could be revealed, and the resource overhead was ≈ 100% compared to the unprotected AES implementation. Thus, highly resilient countermeasures with lower resource overhead to mitigate RPA attacks must be investigated to protect cryptographic algorithms running on FPGAs.

In RPA attacks, TDC sensors use a periodical signal (sample clock), which is in the MHz frequency range and higher than the cipher circuit clock frequency (a 96 MHz sampling clock was used, and the cryptographic circuit executed at 6 MHz in [14]). The sample clock travels through the TDC sensor to detect voltage fluctuations of the cryptographic circuit (see Figure 1-(A)). When the sample clock travels through the Carry elements of the TDC sensor, the sample clock flips the Carry chain output to '1' (bit flips). The number of '1's recorded in the TDC sensor depends on the voltage fluctuations due to the cryptographic circuit in the FPGA. If the cryptographic circuit consumes higher amount of power, the sample clock flips a smaller number of the Carry chain outputs to '1' (see Figure 1-(A) Encryption 1 and Encryption 2). The reason for this observation is that the sample clock does not have enough power in the FPGA PDN to flip Carry chain outputs to '1'. In contrast to this, when the cryptographic circuit consumes a small amount of power, the sample clock flips a higher number of the Carry chain outputs to '1' (see Figure 1-(A) Encryption 3). By analyzing the number of '1's in the Carry chain (TDC sensor reading), the authors in [14] showed the power dissipation of the cryptographic circuit can be deduced.

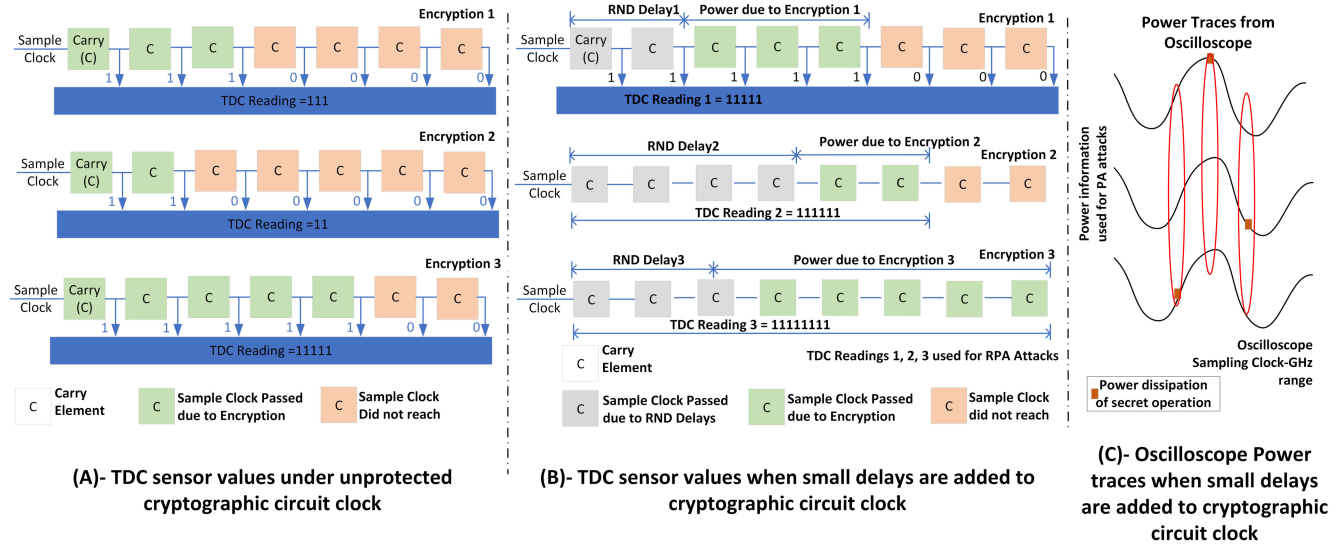
Figure 1-(B) shows when small clock delays are present in the cryptographic clock (RND Delay 1, RND Delay 2 and RND Delay 3 during Encryption 1, Encryption 2 and Encryption3, respectively). Because of the clock delays, the cryptographic clock starts evaluations (register updates) are delayed by the inserted delay time (delays are in picosecond ranges- as shown in Figure 1-(B) and termed RND Delay). During RND Delay, the sample clock of the TDC sensor travels further in the Carry chain flipping outputs to '1'. The number of '1' occurring in the Carry chain due to RND Delays depend on the amount of delay. RND Delay 2 is higher than RND Delay 1; thus, 4-bits are flipped in Encryption 2 in Figure 1-(B) compared to Encryption 1 in Figure 1-(B). The number of '1' of Carry chain outputs due to RND Delays will also be presented in the TDC sensor readings. RND Delays are randomly generated, therefore the TDC sensor readings will get randomly influenced by RND Delays, which acts as noise in the TDC sensor readings. By analyzing the number of '1's in the Carry chain (TDC sensor reading), the adversary gets less information regarding the voltage fluctuations of the cryptographic circuit, hence the TDC sensor readings contain Carry chain outputs due to RND Delays as well as the cryptographic circuit. This increases the noise in the measurement, which reduces the SNR [13].

Figure 1-(C) depicts the effects of random delays for conventional PA attacks. Conventional PA attacks use oscilloscopes with higher sample rates (often in GHz range) to quantize the power dissipations of cryptographic circuit. These delays can be up to two times the execution time of the unprotected circuit [25], [26]. Therefore, the clock delays will misalign the power dissipation of the secret operations, which mitigates the PA attack vulnerabilities [13], [25].

In this paper, we propose *UCloD* which adds fine-tuned delays (in the picosecond (ps) time scale) into the cryptographic circuit clock to mitigate RPA attacks. As explained in our motivational Figure 1, in *UCloD*, using a random number generator, the clock delay added to each clock period is changed. Thus, TDC sensor reading due to the clock delay is randomised, which acts as noise. To examine the efficacy of adding random noise to the TDC measured values, we have mathematically shown that the correlation coefficients are reduced as these minute delays are increased. We confirm this reduction on synthetic traces with added white noise, before showing that an attacker is significantly thwarted when implemented on commercial FPGAs with real data for at least up to a million traces. We also carry out RO-based delay sensor attacks (similar to the attack proposed in [16]) to test the efficiency of the proposed *UCloD* against RO-based RPA attacks. The clock delay-based countermeasures proposed in the literature (which we have discussed in the related work section) to prevent PA attacks, use much higher delays to misalign the power dissipation of the secret operation (Figure 1-(B)). We use very small delays (the smallest delay generate in *UCloD* is 2.1 ps) in *UCloD* to mitigate RPA attacks.

The proposed *UCloD* methodology is implemented on the Lattice Semiconductor iCE40 FPGA (referred to as *UCloD-Lattice*) using the Fine Delay Adjustment mode or Fine Delay Adjustment Feedback mode in the PLL (sysCLOCKPLL) and on the Xilinx Kintex-7 FPGA (referred to as *UCloD-Xilinx*) using IDELAYE2 primitive to test the efficacy of mitigating RPA attacks. Correlation power analysis (CPA) attacks, which use the Pearson correlation coefficient to distinguish the secret key, were carried out to show the effectiveness of *UCloD* clock delay methodology on Lattice Semiconductor and Xilinx FPGA architectures.

The rest of the paper is organized as follows. Section II presents the related work, the previously proposed countermeasures to mitigate RPA attacks and PA attacks. The background is presented in Section III. The proposed *UCloD* methodology, *UCloD* on modern FPGA architectures and the effects of adding clock delays to mitigate RPA attacks are presented in Section IV. *UCloD* implementations on Lattice


Encryption 1
Sample Clock
Carry (C)
C
1 1 1 0 0 0 0
TDC Reading =111
Encryption 2
1 1 0 0 0 0 0
TDC Reading =11
Encryption 3
1 1 1 1 1 0 0
TDC Reading =11111
Carry Element
Sample Clock Passed due to Encryption
Sample Clock Did not reach
RND Delay1
Power due to Encryption 1
1 1 1 1 1 0 0 0
TDC Reading 1 = 11111
RND Delay2
Power due to Encryption 2
TDC Reading 2 = 111111
RND Delay3
Power due to Encryption 3
TDC Reading 3 = 11111111
TDC Readings 1, 2, 3 used for RPA Attacks
Sample Clock Passed due to RND Delays
Sample Clock Passed due to Encryption
Sample Clock did not reach
Power Traces from Oscilloscope
Power information used for PA attacks
Oscilloscope Sampling Clock-GHz range
Power dissipation of secret operation

(A)- TDC sensor values under unprotected cryptographic circuit clock

(B)- TDC sensor values when small delays are added to cryptographic circuit clock

(C)- Oscilloscope Power traces when small delays are added to cryptographic circuit clock

FIGURE 1. Motivation figure.

Semiconductor iCE40 FPGAs and Xilinx Kintex-7 FPGAs as case studies are explained in Section V. The experimental setup is explained in Section VI. The results are presented in Section VII. The discussion and the future work is presented in Section VIII. The paper is concluded in Section IX.

II. RELATED WORK

This section discusses the state of the art RPA attacks and countermeasures against RPA attacks as well as the random clock delay countermeasures proposed to mitigate PA attacks in the literature.

Zick *et al.* [15] proposed the use of delay sensors to detect voltage fluctuations occurring in the FPGAs to detect when the FPGA is under attack. Schellenberg *et al.* [14] proposed first RPA attack using TDC sensors to deduce the secret key of an AES circuit. Gravellier *et al.* [16] proposed to use RO based sensors to detect voltage fluctuations.

The only countermeasure which has been proposed to mitigate TDC sensor-based RPA attacks is ‘Active Fences’ by Krautter *et al.* [24], where a series of ROs are activated randomly, which results in random power fluctuations in the PDN. Thus, the random power fluctuations which occur due to ROs, add noise into TDC sensor readings proportional to the power dissipation of the AES operations. Ergo, a large number of TDC traces were required when compared to the time during which ROs were inactive. According to the authors in [24], ‘Active Fences’ revealed the secret key byte after 300,000 encryptions.

Addition of clock delays to mitigate PA attacks was widely studied in the literature with solutions involving the timing of execution of the cryptographic circuit using a set of random frequencies or a random delayed clock. Güneysu *et al.* [27] proposed a clock multiplexer based clock delay countermeasure using eight phase-shifted clocks with a series of clock multiplexers. The frequency of each phase-shifted clock is identical. Using a random number generator, one phase-shifted clock was chosen as the output to drive the cryptographic circuit. Ravi *et al.* [28] proposed an improved implementation of the clock delay countermeasure proposed in [27] with a floating mean random number generator [29]. Lu *et al.* [30] proposed a random delay execution methodology for FPGAs using random D flip-flops [31] which delay the outputs of D flip-flops at random times using a tapped delay line constructed using general logic. Executing cryptographic circuits using a random frequency clock was also proposed to mitigate PA attacks. The random clock frequency countermeasure proposed in [32] used four clock signals with different clock frequencies and a clock multiplexer. One clock out of four clock signals was used to drive the cryptographic circuit using a random clock frequency. Jayasinghe *et al.* [25] used a dynamic reconfiguration of Xilinx clock managers (referred to as MMCMs) [33] to generate up to three clock signals. Out of three clock signals, one clock signal was randomly chosen in each clock cycle to drive the cryptographic circuit using a random number generator.

The main challenges we try to address in our paper can be summarized as follows.

A. THE MAIN CHALLENGES

- develop a more resilient and extensible countermeasure with low time and area overhead to mitigate RPA attacks
- creating a mathematical model to understand the correlation between the introduced delay to the clock and the Pearson correlation coefficient

Except for the ‘Active Fences’ countermeasure, none of the above countermeasures ([27], [28], [32] and [25]) has shown to be effective against RPA attacks. When compared to the ‘Active Fences’ countermeasure, *UCloD* adds clock delays

TABLE 1. Comparison of the number of delays and the minimal delay size of clock delay and clock randomizing countermeasures proposed in the state of the art related work.

Countermeasure	Delays Generated via	Smallest Delay Demonstrated	Number of Delays Demonstrated
Guneysu et al. [27]	Phase shifted Clocks	1/8 clock period=5.20ns∓	15 *
Ravi et al. [28]	Phase shifted Clocks	1/4 clock period=10.4ns∓	39 *
Lu et al. [30]	TDL	1 delay element=**ND**	**ND**
Fritzke et al. [32]	BUFGMUX	**ND**	83
Jayasinghe et al. [25]	MMCM	5ps∓	up to 3,072
UCloD (This paper)	TDL	78ps	160

∓ Calculated based on the information presented in the manuscripts
ND– Not Disclosed
∗ According to authors of [25]

to induce noise in the TDC sensor readings and demonstrates CPA attack resistance for at least up to one million encryptions. Compared to [27] and [28], *UCloD* can achieve higher number of distinct clock delays generated using tapped delay modules. We demonstrate up to 160 distinct clock delays in *UCloD* using cascaded tapped delay lines compared to ≈15 delays in [27] and ≈39 delays in [28]. Compared to random delay countermeasure proposed by Lu *et al.* [30], *UCLOD* induces PVT invariant delays using the hardware components which are fabricated into the FPGAs. Compared to [32], *UCloD* adds clock delays rather than switching the clock frequency where changing the clock frequency will affect the critical path. The work proposed in [25] has a time overhead of 1.7× and an area overhead of 1.3×, and requires around $34\mu s$ for the reconfiguration of the MMCM whenever a new set of frequencies are generated. The tapped delay lines used in *UCloD* can be reconfigured without an overhead to generate differing delays at run time.

B. CONTRIBUTIONS

- For the first time, a novel clock delay methodology on FPGAs which has minimal area and time overheads and is resistant to RPA attacks for up to one million encryptions (the state of the art countermeasures was only resistant up to 300,000 encryptions).
- We show how the clock delays can be achieved by the use of TDLs fabricated in the FPGAs.
- We mathematically model the Pearson correlation coefficient reduction when the clock delays are added into cryptographic circuits and verify the model using simulations.
- We demonstrate two *UCloD* implementations on Xilinx 7 Series and Lattice Semiconductor iCE40 FPGA architectures as a case study.

To the best of our knowledge, this is the first robust clock delay countermeasure proposed to mitigate RPAs.

III. BACKGROUND

This section briefly outlines necessary background information about AES block cipher algorithm, CPA attacks and TDC sensors.

A. ADVANCED ENCRYPTION STANDARD (AES)

The Advanced Encryption Standard (AES) is the most widely used block cipher algorithm, which is standardized under Federal Information Processing Standard, FIPS-197 [7]. AES algorithm performs four operations (SubByte, ShiftRow, MixColumn and AddRoundKey operations) in each round (except the initial and the last round) to generate the ciphertext. The number of rounds is varied from 10, 12 and 14 for 128-bit, 192-bit and 256-bit secret key, respectively. In this paper, we have used AES 128-bit implementation (similar to the AES circuit used in [14]) to test the CPA attack resistance of *UCloD* implementations.

B. CORRELATION POWER ANALYSIS– CPA ATTACKS

Correlation power analysis (CPA) attack is one of the most powerful side channel analysis attacks proposed in the literature. CPA attacks use the Pearson correlation coefficient [34] to estimate the correlation between the measured power and the anticipated power consumption based on a guessed key. The Pearson correlation coefficient (r) [35] between variable X and Y can be expressed as shown in Equation 1.

$$r(X,Y) = \frac{COV(X,Y)}{\sigma_X \sigma_Y}, \tag{1}$$

where σ_X and σ_Y are the standard deviation of X and Y, respectively. $COV(X, Y)$ is the covariance between X and Y. In a CPA attack, one variable (say X) is the power dissipation of an execution of the cryptographic algorithm (say Y). The power dissipation of the circuit is often measured by an oscilloscope and the other variable is the hypothetical power [13]. The hypothetical power is calculated based on a guessed key using a power model. Block cipher algorithm implementations, such as AES circuits and AES software implementations, are vulnerable to CPA attacks due to input (data) dependent power dissipations of CMOS logic gates [13]. These power dissipations are quantized by an oscilloscope or a sensor such as, TDC or RO-based voltage sensor. Authors in [14] and [16] have shown how AES is vulnerable to RPA attack using TDC and RO sensors. Then the adversary builds hypothetical power dissipation, such as hamming weight (counting number of '1's) or hamming distance (counting the number of bit flips), to model what circuit might consume based on input/ output using a guessed key. The guessed key with highest correlation coefficient corresponds to the secret key used in the cryptographic device.

C. TIME TO DIGITAL CONVERTER – TDC TO DETECT VOLTAGE FLUCTUATIONS

Time to Digital Converter (TDC) is a sensor which measures the delay between two signals and is commonly fabricated on ASICs or ICs [36]. On FPGAs, TDC sensors are implemented as a custom hardware design using carry propagation logic in the FPGA which is precisely routed to achieve minimum latency in adders. The TDC sensor proposed in [14], [15] to deduce voltage fluctuations in Xilinx FPGAs used the Xilinx CARRY4 chain [37]. A clock signal is supplied as the input to the TDC sensor. The input clock signal will propagate through the CARRY4 chain and a series of latches. Due to voltage fluctuations, input clock signal travelling through the CARRY4 chain will get delayed, compared to the input clock signal sampling the carry chain values into the latches. Calculating how far the input clock travelled in the TDC sensor will reveal side channel information regarding the energy consumption (power dissipation) of the other circuits (such as cryptographic circuits) executing on the FPGA, simultaneously with the TDC sensor. The voltage fluctuations which occur in the PDN are inversely proportional to the readings of the TDC sensor (higher the voltage fluctuation which occurs in the PDN, lower the TDC reading). A pre-processing algorithm, such as bubble error correction [38], is often applied to the readings from the TDC sensor before performing CPA attacks to make the TDC readings more accurate. The readers are advised to refer to [14], [15] for more information regarding TDC sensors.

D. RO-BASED VOLTAGE SENSORS TO DETECT VOLTAGE FLUCTUATIONS

In this subsection, we briefly explain RO-based voltage fluctuation sensors to perform RPA attacks. The RO-based sensor proposed by Gravellier *et al.* [16] is shown in Figure 2 which consists of an RO, built using LUTs (FPGA look-up tables) and a counter (such as a Johnson Counter [39]– referred to as JRC) to count the number of clock cycles recorded by the RO (shown in Figure 2). Due to voltage fluctuations, the frequency of the RO will be reduced momentarily. This frequency reduction will be reflected in the JRC counter value (see Figure 2). The frequency change in RO is captured by the JRC counter value. The sample clock (as shown in Figure 2) samples the JRC counter values and copies the values into memory. According to the authors of [16], multiple RO-based sensors are required to sense voltage fluctuations occurring in the FPGA to carry out a successful RPA attack.

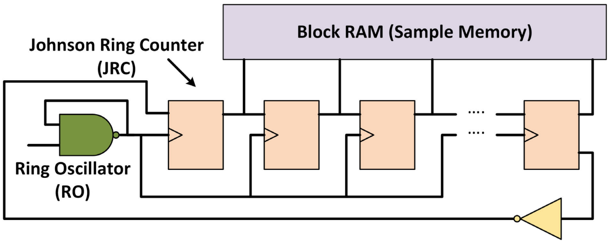


FIGURE 2. RO sensor proposed in [16].

IV. UCloD-CLOCK DELAY METHODOLOGY

This section presents *UCloD*, the proposed clock delay methodology to mitigate RPA attack vulnerabilities, *UCloD* implementation on modern FPGA architectures and the effects of adding delays on TDC sensors and CPA attacks. *UCloD* uses tapped delay lines fabricated in the FPGA to induce clock delays (which are less than a clock period of the circuit under attack) to cryptographic circuit clock. Therefore, the voltage fluctuations which occur in the PDN due to cryptographic operations occur after random time intervals. Thus, the TDC sensor measurements/readings are randomized which will be used to mitigate RPA attacks.

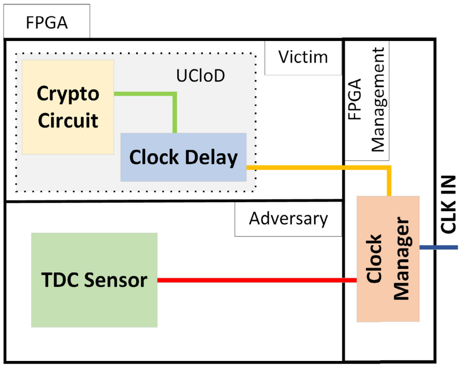


FIGURE 3. The proposed *UCloD* methodology.

Figure 3 depicts the architecture of the proposed *UCloD* methodology. In a multi-tenant FPGA architecture, the clock input from a clock source (typically an oscillator) is sent through a clock manager (such as PLL or MMCM module) to generate the clock frequencies required by different users. In the proposed *UCloD* methodology, the generated clock signal for the cryptographic device is sent through the clock delay generator. The clock delay generator delays each clock period by $n\delta$ seconds (δ is typically in either nanosecond or picosecond range - typically 2ps to 150ps depending on the FPGA architecture), n is generated randomly, using a random number generator (compared to the sampling clock, $n\delta$ delay value is much smaller).

The delay generator is a programmable tapped delay line, and the delay amount of the output clock can be changed in each clock cycle. The tapped delay line has a finite number of delay lines (which we denoted as N) which are connected in series. Let us denote the propagation of a delay element as δ seconds. We can choose from N possible delay lines, ranging from 0 to $(N-1)\delta$ seconds, which will result in a clock delay between 0 to $(N-1)\delta$ seconds. Tapped delay lines can be cascaded (connected in series) to generate larger delays. The total delay output from the cascaded tapped delay line is the sum of delays in the delay line. Using a random number generator, a random delay output is selected (between 0 and $N-1$) from the delay generator; thus, the output clock is also delayed by a random delay varying between 0s to $(N-1)\delta$ seconds. We refer to *UCloD* implementations by the number of delay lines. E.g., a *UCloD* implementation with x number of tapped delay lines is denoted as *UCloD(N=x)*.

A. UCloD ON MODERN FPGA ARCHITECTURES

This section explains how to implement *UCloD* methodology in current FPGA architectures to generate random clock delays. Tapped delay lines are often fabricated as hardware modules in the FPGA; they are often used to phase shift the clock in clock management modules (such as PLLs) and remove the effects of path delays in I/O data [40]. We repurpose this delay line to shift the clock ever so slightly, to defeat RPA attacks. Figure 4 shows the architecture of a typical tapped delay module which is fabricated in FPGAs. The input signal is passed through a set of delay elements which adds a precise propagation delay into the signal from each delay element. Using a multiplexer, output from one delay element is chosen. In *UCloD*, after passing through the n^{th} $(0 \leq n \leq N-1)$ delay element in the tapped delay line, the clock signal gets delayed by $n\delta$ seconds.

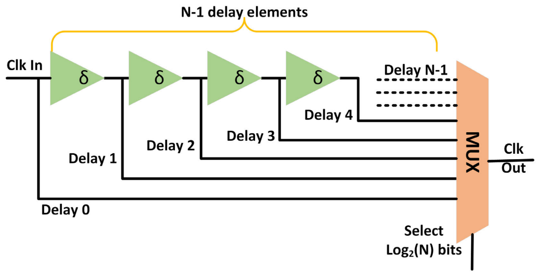


FIGURE 4. Architecture of a generic tapped delay line fabricated in FPGAs.

By choosing one output from the tapped delay line (by varying number of taps in the clock signal) using a random number generator, the clock delays in the output clock are randomly generated.Thus, the n^{th} $(0 \leq n \leq N-1)$ delay element in the tapped delay line delays the clock signal by $n\delta$ seconds.

Table 2 tabulates the clock capable tapped delay lines in different FPGA architectures (in some FPGAs, clock signals cannot be routed through the tapped delay lines). The first column shows the FPGA manufacturer. The FPGA architecture name, the FPGA primitive in which the tapped delay line is fabricated, the location of the FPGA primitive, the size of the tapped delay line and the propagation delay of each delay element are shown in column two, three, four, five and six, respectively. Tapped delay lines in I/O locations are often capable of being cascaded to create larger tapped delay lines to generate a large range of delays depending on the FPGA architecture.

B. ADDING CLOCK DELAYS TO CRYPTOGRAPHIC CIRCUITS AND THE EFFECT ON TDC SENSOR READINGS

This section describes how the adding of clock delays to the cryptographic circuit affects the TDC sensor readings and the consequences of the affected readings on CPA attacks. Furthermore, we show mathematically and through simulations with synthetic traces that the Pearson correlation coefficient of *UCloD* implementation will reduce significantly when compared to unprotected implementation. Finally, we show that the Success Rates of obtaining the secret keys reduces in real FPGA implementations.

TABLE 2. TDLs in modern FPGA architectures.

FPGA Vendor	FPGA Architecture	Primitive	Loc.	Tapped Delay Size(N)	δ (ps)
Xilinx	5 Series	IODELAY	I/O	64	78
	6 Series	IODELAY1	I/O	256	78
	7 Series	IDELAYE2	I/O	32	78
	Ultra Scale	IDELAYE3 ODELAYE3	I/O	512	2.1-12
Lattice	iCE40	sysCLOCKPLL	PLL	16	150
Microsemi	PolarFire	PLL	PLL	256	25
Intel Altera	Stratix V Stratix IV Arria V Arria II Cyclone 10 LP Cyclone V Cyclone IV	ALTIOBUF	I/O	16	50-60

As discussed in the Section III-C, the readings from the TDC sensors can be used without converting to power dissipation values in order to conduct CPAs. Let us assume, without the clock delay, the clock signal in the TDC travels up to P_E bit in the TDC sensor (Figure 5-A). With the clock delay, the AES circuit starts evaluating late (after $n\delta$ seconds, to be precise) compared to identical unprotected TDC sensor operation. Therefore, the voltage fluctuations in the PDN happen $n\delta$ seconds after adversary's clock signal starts propagating through the TDC sensor. Thus, the clock signal travels further in the TDC sensor before being sampled by the sampling clock in the TDC sensor, which we refer to as P_O (as shown in the Figure 5-B). $P_O \geq P_E$ because the delay value is greater than or equal to 0. When the delay value is 0, $P_O = P_E$.

Let us assume that the propagation time of a single element (each Carry element in Figure 5-(B)) in the TDC sensor is Γ. Δ can be calculated according to Equation 2. By randomizing n, the Δ which occurs in the TDC sensor will also be randomized.

$$\Delta = \frac{n\delta}{\Gamma} \tag{2}$$

According to the Equation 1, the Pearson correlation coefficient (r) between TDC sensor values P_E and the hypothetical power H for the correct key candidate can be represented as shown in Equation 3, where σ_{P_E} and σ_H are the standard deviation of P_E and H, respectively. $COV(P_E, H)$ is the covariance between P_E and H. We name each Pearson correlation coefficient based on the variables used in the calculation (e.g., Pearson correlation coefficient among P_E

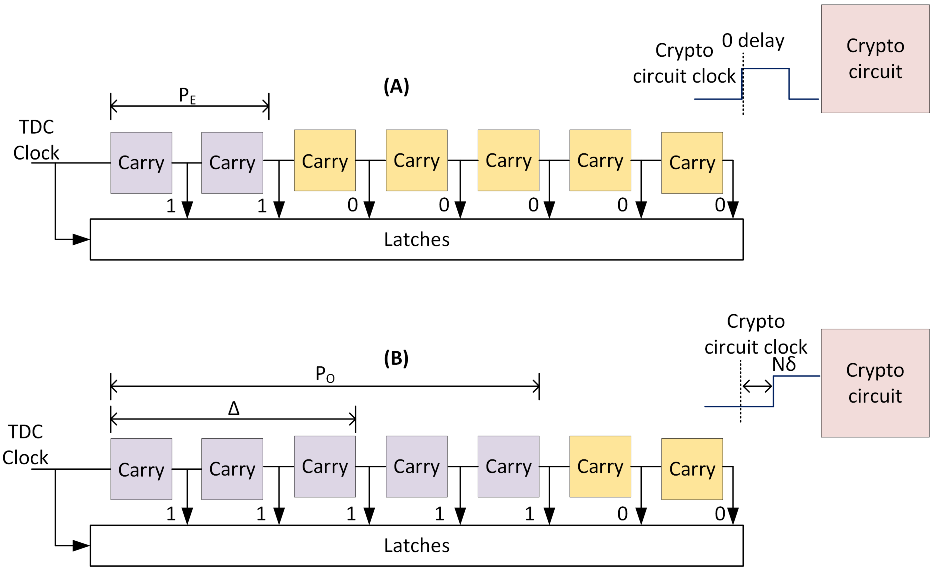


FIGURE 5. (A)- TDC sensor operation under unprotected clock; (B)- TDC sensor operation under *UCloD* methodology (with clock delay).

and H is referred to as $CPA(P_E, H)$).

$$CPA(P_E,H) = \frac{COV(P_E,H)}{\sigma_{P_E}\sigma_H} = \frac{E(P_E.H)-E(P_E)E(H)}{\sigma_{P_E}\sigma_H} \quad (3)$$

When *UCloD* countermeasure is active, due to clock delays, voltage fluctuations recorded by the TDC sensor is P_O. Therefore, the Pearson correlation coefficient among P_O and H can be represented as shown in the Equation 4.

$$CPA(P_O,H) = \frac{COV(P_O,H)}{\sigma_{P_O}\sigma_H} = \frac{E(P_O.H)-E(P_O)E(H)}{\sigma_{P_O}\sigma_H} \quad (4)$$

In ideal TDC sensor operation, P_O is the sum of P_E and Δ (P_E, due to voltage fluctuations which occur in the PDN by the cryptographic circuit, and Δ due to the clock delay in the cryptographic circuit). Therefore, P_O can be represented as Equation 5.

$$P_O = P_E + \Delta \quad (5)$$

Since Δ is due to clock delay, Equation 4 can be rewritten as shown in Equation 6.

$$CPA(P_O,H) = \frac{COV((P_E+\Delta),H)}{\sigma_{(P_E+\Delta)}\sigma_H} = \frac{E(P_E.H)+E(\Delta.H)-E(P_E)E(H)-E(\Delta)E(H)}{\sigma_{(P_E+\Delta)}\sigma_H} \quad (6)$$

H and Δ are independent. Therefore, $E(H.\Delta)$ can be expressed as shown in the Equation 7.

$$E(H.\Delta) = E(H)E(\Delta) \quad (7)$$

By substituting Equation 7 into Equation 6, Equation 6 can be simplified as shown in Equation 8.

$$CPA(P_O,H) = CPA(P_E,H) \times \frac{\sigma(P_E)}{\sigma(P_E+\Delta)} \quad (8)$$

Since P_E and Δ are independent, the variance of $P_E + \Delta$ is the variance of P_E + variance of Δ, which is shown in Equation 9.

$$\frac{\sigma(P_E)}{\sigma(P_E+\Delta)} = \sqrt{\frac{\sigma(P_E)^2}{\sigma(P_E+\Delta)^2}} = \sqrt{\frac{\sigma(P_E)^2}{\sigma(P_E)^2+\sigma(\Delta)^2}} \quad (9)$$

Equation 8 can be simplified by substituting Equation 9 into the Equation 8, as shown in Equation 10.

$$CPA(P_O,H) = CPA(P_E,H) \times \sqrt{\frac{\sigma(P_E)^2}{\sigma(P_E)^2+\sigma(\Delta)^2}} \quad (10)$$

As shown in Equation 10, the Pearson correlation coefficient, when the *UCloD* is active, is reduced by the factor of $\sqrt{\frac{\sigma(P_E)^2}{\sigma(P_E)^2+\sigma(\Delta)^2}}$ which is ≤ 1. When $\Delta > 0$ and sufficiently random, the Pearson correlation coefficient of a *UCloD* implementation is reduced when compared to the Pearson correlation coefficient of the unprotected implementation.

To test the effect of adding clock delays to cryptographic circuits, we ran CPA attacks on simulated TDC sensor readings for AES block cipher execution using MATLAB with 0.5dB additive white Gaussian noise (using 'awgn' command in MATLAB). We set $\Gamma = 1$ and $\delta = 1$ for the simplicity of the simulations which will lead to $\Delta = n$. Thus, for each TDC sensor simulation, we added a random TDC reading Δ which is between 0 and $N - 1$. N is incremented from 16 to 80. The Pearson correlation coefficients are plotted in Figure 6. The Pearson correlation coefficients of the unprotected AES, *UCloD(N = 16)*, *UCloD(N = 32)*, *UCloD(N = 48)*,

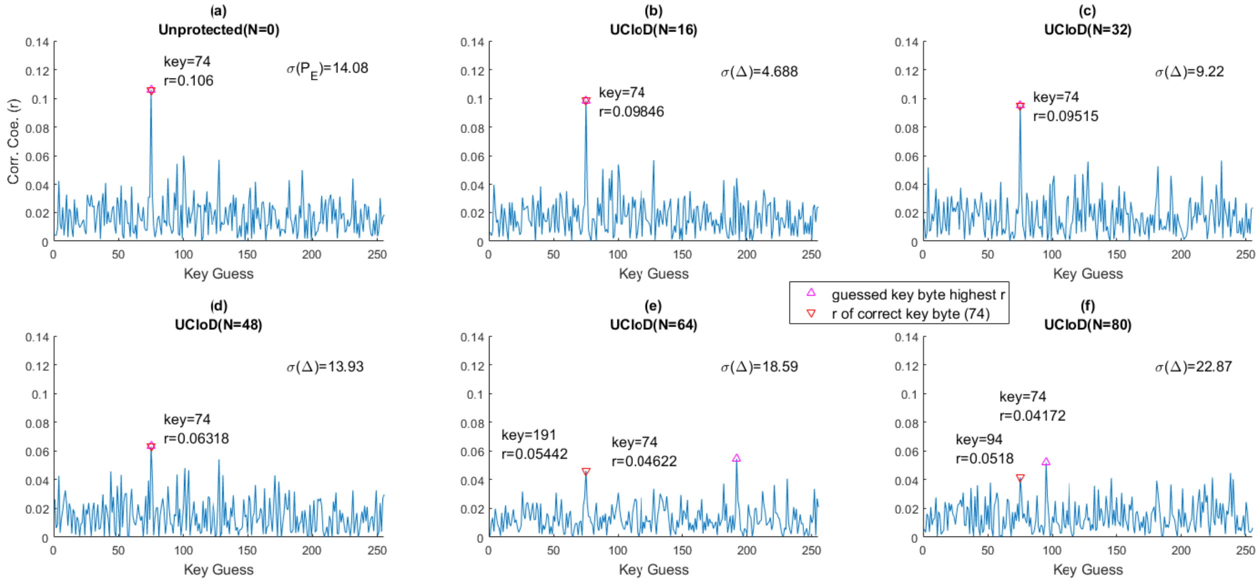


FIGURE 6. CPA attacks on simulated unprotected AES and *UCloD*.

TABLE 3. Comparison of pearson correlation coefficient (r) from the simulated TDC traces vs. using the equation 10.

	Unprot.	*UCloD* (N=16)	*UCloD* (N=32)	*UCloD* (N=48)	*UCloD* (N=64)	*UCloD* (N=80)
r calculated from TDC traces	0.1060	0.0984	0.0951	0.0631	0.0462	0.0417
σ	$14.08^{\dagger}$	4.68*	9.22*	13.93*	18.59*	22.87*
r calculated from Equation 10	–	0.1011	0.0901	0.0774	0.0663	0.0578

† denotes $\sigma(P_E)$; * denotes $\sigma(\Delta)$

UCloD($N = 64$) and *UCloD*($N = 80$) simulations are shown in the Figure 6-(a), Figure 6-(b), Figure 6-(c), Figure 6-(d), Figure 6-(e) and Figure 6-(f), respectively. The secret key byte used in the AES simulation was '74'. The Pearson correlation coefficient of the secret key byte ('74') is annotated as ▽ and the key byte with highest correlation coefficient (r) of simulation is annotated as △ in the Figure 6. According to the Figure 6, Pearson correlation coefficients of the secret key byte ('74') are reduced when the random delay range is increased. Another random key byte (not the secret key byte) records the highest Pearson correlation coefficient which fails the CPA attack. Using the simulated TDC sensor data, we estimated the Pearson correlation coefficient of the secret key byte ('74') using the derived Equation 10. The estimated values of r for *UCloD* simulations are tabulated in the Table 3. The differences between r calculated from simulated TDC sensor readings and r calculated using Equation 10 are due to the fact that H and Δ are not fully independent ($E(H.\Delta) \approx E(H)E(\Delta)$). The maximum error between the actual correlation coefficient and the predicted correlation coefficient of the secret key byte is 0.020 for the simulations we conducted.

V. CASE STUDIES: *UCloD* ON XILINX AND LATTICE SEMICONDUCTOR FPGAs

We implemented *UCloD* on Xilinx 7 series FPGA architecture (referred to as *UCloD-Xilinx*) and Lattice Semiconductor iCE40 FPGA architecture (referred to as *UCloD-Lattice*).

A. UCloD-XILINX *IMPLEMENTATION*

UCloD-Xilinx was implemented using Xilinx IDELAYE2 primitives, where each IDELAYE2 primitive can add 31 distinct delays into the clock output, according to the Table 2. Xilinx IDELAYE2 primitives can be cascaded to create large tapped delay lines. Xilinx IDELAYE2 primitives need Xilinx IDELAYCTRL primitive [41] instantiated with a reference clock. During start-up, Xilinx IDELAYE2 primitives will be calibrated by Xilinx IDELAYCTRL primitive to generate a process, voltage and temperature (PVT) invariant delay [37].

B. UCloD-LATTICE *IMPLEMENTATION*

UCloD-Lattice implementation uses the tapped delay line in the PLL (sysCLOCKPLL module) of the Lattice Semiconductor FPGA to add clock delays. The tapped delay line in PLL can induce delays up to 2.4ns according to the Table 4. Lattice Semiconductor FPGAs have at most two PLLs. *UCloD* was implemented using one PLL (running in Fine Delay Adjustment–FDA mode [42]) to generate clock delays for the cryptographic circuit and the other PLL supplies the clock to operate the TDC sensor. Because of the availability of a single PLL, we only use one tapped delay line to demonstrate the effectiveness of *UCloD-Lattice* implementation.

VI. EXPERIMENTAL SETUP

This section describes the experimental setup. We implemented *UCloD-Xilinx* on a SASEBO GIII [43] FPGA board which has a Xilinx Kintex-7 325T FPGA. The total number of IDELAYE2 primitives available in the Xilinx Kintex-7 325T FPGA is 500. An FTDI RS232 module was used to transfer TDC sensor data to the PC. We varied cascaded IDELAYE2 modules from one to five to create a large tapped delay line (tapped delay length (N) will vary from 32 to 160). The architecture of *UCloD-Xilinx* implementation on SASEBO GIII board is shown in Figure 7 with cascaded IDELAYE2 primitives. Each *UCloD-Xilinx* implementation was named based on N. The tapped delay lines in IDELAYE2 modules and the TDC sensor are expanded in Figure 7. *UCloD-Xilinx* implementation has one IDELAYE2 primitive, two IDELAYE2 primitives, three IDELAYE2 primitives, four IDELAYE2 primitives and five IDELAYE2 primitives. Each aforementioned *UCloD* implementation was named *UCloD-Xilinx(N = 32)*, *UCloD-Xilinx(N = 64)*, *UCloD-Xilinx(N = 96)*, *UCloD-Xilinx(N = 128)* and *UCloD-Xilinx(N = 160)*, respectively. A 192MHz clock was used for IDELAYCTRL module to calibrate each IDELAYE2 module. The size of the TDC sensor was 160-bits, which was constructed using Xilinx CARRY4 primitives.

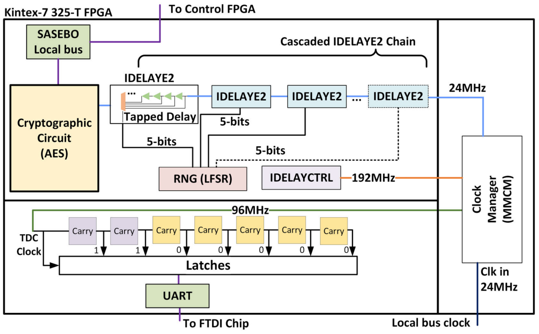


FIGURE 7. ***UCloD-Xilinx* implementation on SASEBO GIII Board (with tapped delay lines in IDELAYE2 modules and TDC sensor expanded in Figures 4 & 5).**

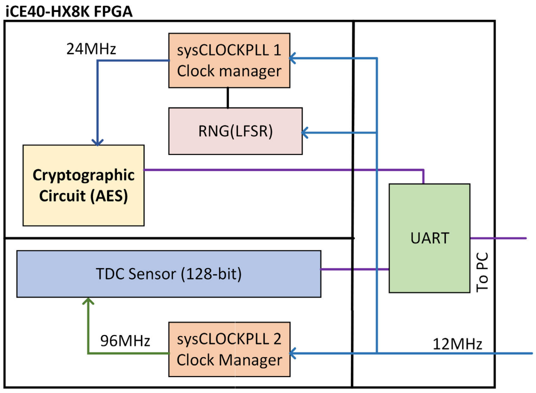

iCE40-HX8K FPGA
24MHz
sysCLOCKPLL 1 Clock manager
RNG(LFSR)
Cryptographic Circuit (AES)
UART
To PC
TDC Sensor (128-bit)
96MHz
sysCLOCKPLL 2 Clock Manager
12MHz

FIGURE 8. ***UCloD-Lattice* implementation on lattice semiconductor iCE40 FPGA Board.**

We implemented *UCloD-Lattice* on Lattice Semiconductor iCE40-HX8K FPGA board which we referred to as *UCloD-Lattice(N = 16)* and is shown in Figure 8. There is only one tapped delay line available on the iCE40-HX8K FPGA to generate clock delays as explained in the Section V-B. The inbuilt FTDI chip in iCE40-HX8K FPGA board was used to transfer TDC data and ciphertext. Thus, only one *UCloD-Lattice* implementation was used to test the CPA attack resistance of *UCloD* on the Lattice Semiconductor iCE40-HX8K FPGA board. The size of the TDC sensor was 128-bits, which was constructed using SB_CARRY primitives. We used the AES circuit discussed in [14], [24], [44], which takes 50 clock cycles to produce ciphertext. Identical AES circuits were used on the SASEBO GIII board and the Lattice Semiconductor iCE40-HX8K FPGA board to test the CPA attack vulnerabilities of the unprotected AES and the resistance of *UCloD* implementations. Each CPA attack was repeated 100 times to calculate the Success Rates [45] to calculate the average number of TDC sensor traces (readings) needed to reveal the secret key. Key byte 0 was chosen to calculate Success Rates ([14], [24] also showed results for a single key byte, thus comparisons can be easily made).

The floor plan of the *UCloD-Xilinx* implementation is shown in Figure 9. The TDC sensor and the AES circuit are placed such that there is a distance between the components. The IDELAYE2 and IDELAYCTRL modules are placed in a non-restricted manner (we let Xilinx ISE tool decide the best locations to meet timing constraints).

RO sensor-based RPA attacks are carried out using 128 RO sensors. We let Xilinx ISE tool decide each location of each RO sensor to match the clock timing constraints for each sensor.

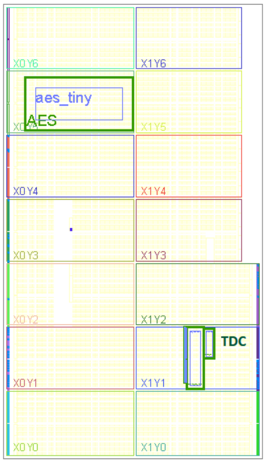
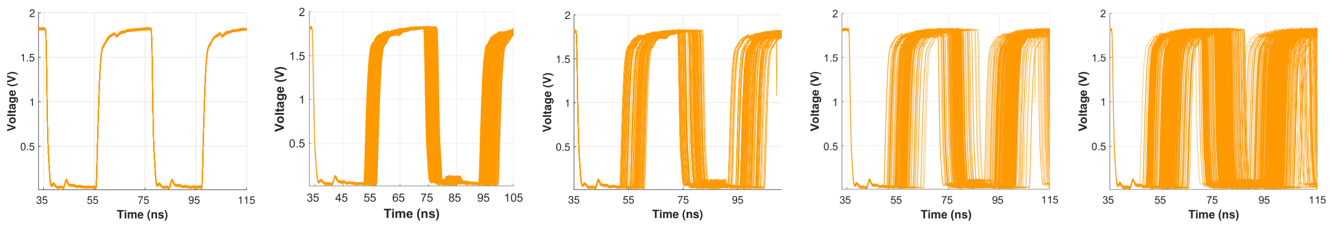


FIGURE 9. **Floor plan of *UCloD-Xilinx* implementation on SASEBO GIII-Xilinx Kintex325T FPGA.**

The AES circuit was executed at 24MHz and the TDC sensor was executed at 96MHz in all the implementations. Thus, the power dissipation of each AES operation was captured by the TDC sensor four times, similar to [14]. The maximum delay generated in *UCloD-Xilinx* implementations



(a) Without Delay -Unprotected

(b) *UCloD-Xilinx(N=32)*

(c) *UCloD-Xilinx(N=64)*

(d) *UCloD-Xilinx(N=96)*

(e) *UCloD-Xilinx(N=128)*

FIGURE 10. Clock delays measured by oscilloscope of AES implementations on *UCloD-Xilinx*.

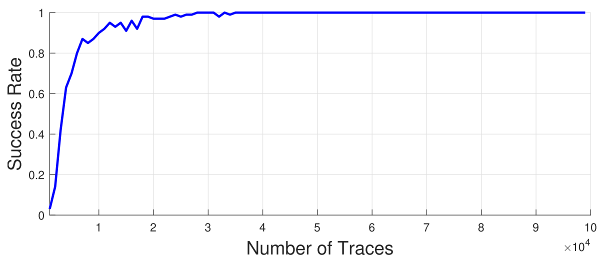


FIGURE 11. CPA attacks based on TDC sensor values measured from unprotected AES encryption running on SASEBO GIII.

TABLE 4. Countermeasures proposed/proven secure against RPA attacks.

Countermeasure	Overhead		CPA
	Resource (%)	Time (%)	Resistance
Active Fences	100	0	300,000
UCloD Xilinx(N=128) (this paper)	Slice LUTs 0.2 Slice Reg.s 4.8	1	1,000,000
UCloD Lattice(N=16) (this paper)	LogicCells 0.5	1	1,000,000

(*UCloD-Xilinx(N = 160)* implementation to test RO-based RPA attacks) was 12.48ns (78ps per delay element × 160 elements). The maximum delay generated in *UCloD-Lattice* implementation was 2.4ns (150ps per delay element × 16 elements). 128-bit linear-feedback shift registers (LFSRs) were used to generate random numbers to execute *UCloD-Xilinx* and *UCloD-Lattice* implementations. When the generated random number is not sufficiently random, a True Random Number Generator (TRNG) or an unbiased random number generation methodology can be used [29].

VII. RESULTS

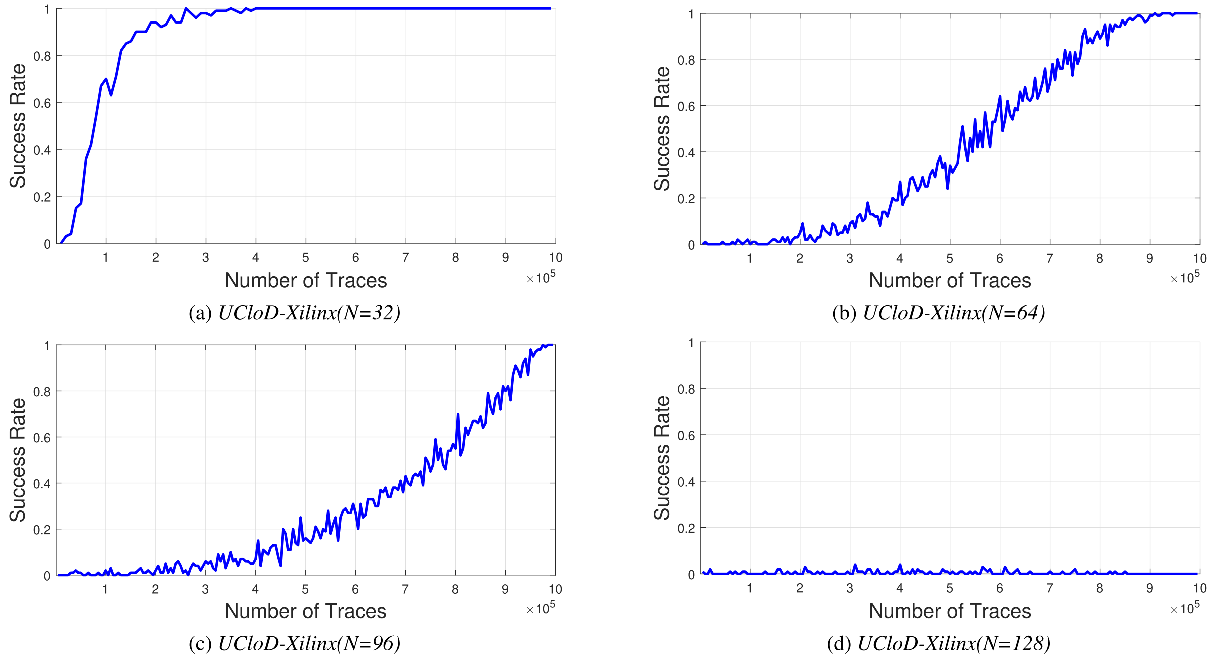
This section presents CPA attack results and the implementation details obtained for *UCloD-Xilinx* and *UCloD-Lattice* implementations. First, we measured the clock delays induced by the IDELAYE2 primitives. Using Keysight DSOS404A oscilloscope, we measured the clock delay ranges and the results are plotted in Figure 10. Figure 10-(a) shows the clock used in the unprotected AES implementation (without *UCloD*) without clock delays. Figure 10-(b), Figure 10-(c), Figure 10-(d) and Figure 10-(e) show the clock delay graph for *UCloD-Xilinx(N = 32)*, *UCloD-Xilinx(N = 64)*, *UCloD-Xilinx(N = 96)* and *UCloD-Xilinx(N = 128)* implementations, respectively. As shown in Figures 10-(b), (c), (d) and (e), when the number of IDELAYE2 primitives are increased, the clock delay range increases. The higher clock delay range results in higher Δ range in the TDC sensor readings.

The CPA attack results for unprotected AES running on the SASEBO-GIII board are shown in Figure 11, demonstrating that the secret key byte is revealed (Success Rate reached 1.0) within 22,000 encryptions.

CPA attack results for *UCloD-Xilinx(N = 32)*, *UCloD-Xilinx(N = 64)*, *UCloD-Xilinx(N = 96)* and *UCloD-Xilinx (N = 128)* implementations are shown in the Figure 12-(a), (b), (c) and (d), respectively.

According to Figure 12-(a), the CPA attacks revealed the secret key byte in around 320,000 encryptions for *UCloD-Xilinx(N = 32)* (when one IDELAYE2 primitive is used) implementation. Figure 12-(b) showed that *UCloD-Xilinx(N = 64)* (when two IDELAYE2 primitives are cascaded) implementation revealed the secret key byte in around 910,000 encryptions. According to Figure 12-(c), *UCloD-Xilinx(N = 96)* (when three IDELAYE2 primitives are cascaded) implementation revealed the secret key byte in around 975,000 encryptions. Figure 12-(d) demonstrated that*UCloD-Xilinx(N = 128)* (when four IDELAYE2 primitives are cascaded) implementation did not reveal the secret key byte for one million encryptions. The maximum Success Rate was 0.05 for *UCloD-Xilinx(N = 128)* implementation.

We carried out the Sliding Window [46] (referred to as SliW) preprocessing method on the TDC traces collected for the *UCloD-Xilinx(N = 128)* implementation (which demonstrated a maximum Success Rate of 0.05 when raw TDC sensor readings are used - Figure 12-(d)). We varied the window size (WSize) between two to six (WSize one is the CPA attack without any preprocessing–Figure 12-(d)) to combine the TDC sensor readings to remove the effects of adding clock delays to the AES circuit. The CPA attack results for SliW preprocessed traces are shown in Figure 13. As shown in Figure 13, WSize = 3 was only able to reach a Success Rate of 0.1, at around 900,000 encryptions. This is the highest Success Rate we could achieve even with preprocessing of the TDC sensor readings. We also tried Principal Compo-



(a) *UCloD-Xilinx(N=32)*

(b) *UCloD-Xilinx(N=64)*

(c) *UCloD-Xilinx(N=96)*

(d) *UCloD-Xilinx(N=128)*

FIGURE 12. CPA attacks on *UCloD-Xlinx* implementations voltage fluctuations measured via TDC sensors.

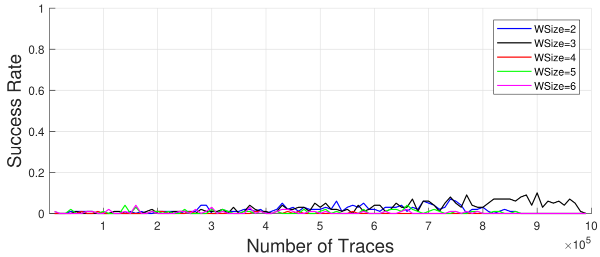


FIGURE 13. CPA attacks on *UCloD-Xilinx(N = 128)* preprocessed traces using sliding window(SliW) preprocessing method.

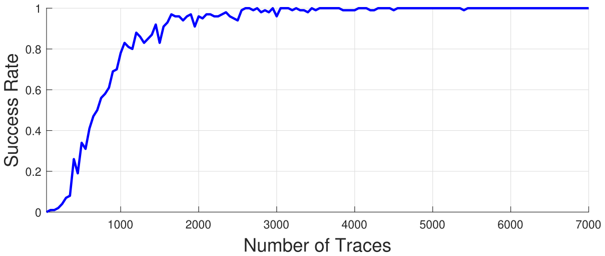


FIGURE 14. CPA attacks based on TDC sensor readings on unprotected AES running on lattice semiconductor iCE40-HX8 FPGA board voltage fluctuations measured via TDC sensors.

nent Analysis [47] preprocessing on the traces from *UCloD-Xilinx(N = 128)* implementation, the CPA attacks could only reach a Success Rate up to 0.02.

CPA attacks were carried out on Lattice Semiconductor iCE40-HX8 FPGA board running unprotected AES, and the results are shown in Figure 14. As shown in Figure 14, the secret key byte was revealed within 2,500 encryptions.

The CPA attack results for *UCloD-Lattice(N = 16)* implementation running on Lattice iCE40-HX8 FPGA board are presented in the Figure 15. As shown in the Figure 15, the *UCloD-Lattice(N = 16)* implementation does not reveal the secret key byte even for one million encryptions. We also carried out SliW preprocessing method on TDC sensor readings acquired for *UCloD-Lattice(N = 16)* implementation. SliW-based CPA attacks could not increase the Success Rates.

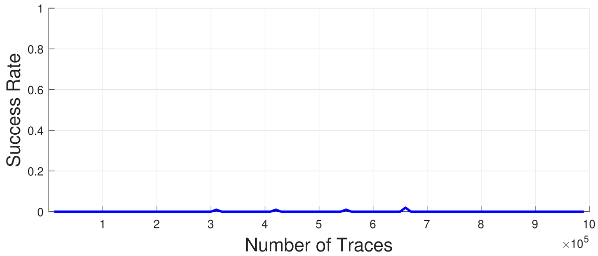


FIGURE 15. CPA attacks based on TDC sensor readings on *UCloD-Lattice(N = 16)* implementation running AES on iCE40-HX8K FPGA board voltage fluctuations measured via TDC sensors.

UCloD-Lattice(N = 16) implementation does not reveal the secret key is because the 128-bit TDC sensor (the maximum possible size was 128bits) which was too small and therefore could not capture the full voltage fluctuation due to the clock delay range. We are planning to investigate further with efficient delay sensors as future work).

Table 4 compares the resource overhead, time overhead and the RPA attack resistance demonstrated (using CPA attacks) of the state of the art countermeasures proposed in the literature to prevent RPA attacks with the *UCloD-Xilinx(N = 128)* implementation and *UCloD-Lattice(N = 16)* implementation. According to Table 4, both *UCloD-Xilinx(N = 128)* and *UCloD-Lattice(N = 16)* implementations are resistant to RPA attacks with resource overheads of less than ≈5%

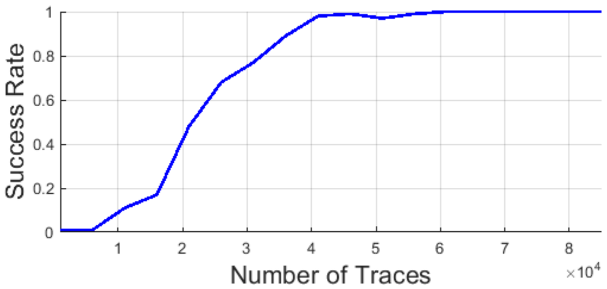


FIGURE 16. CPA attacks on unprotected AES implementation measured using 128 instances of RO-based power sensors.

and time overheads of a half a clock cycle (we delay reading the ciphertext until the negative edge of the clock which adds a half clock cycle overhead), respectively, compared to the corresponding unprotected AES implementations. The added half a clock cycle increases the time overhead by ≈1% (the circuit takes 50 clock cycles to produce the ciphertext as explained in Section VI). The time overheads were measured using a Keysight DSOS404A oscilloscope, the resource utilization of *UCloD-Xilinx(N = 128)* was calculated using Xilinx ISE 14.7 tool, and *UCloD-Lattice(N = 16)* was calculated using Lattice iCEcube2 tool.

A. UCloD AGAINST RO-BASED RPA ATTACKS

To test the efficiency of mitigating RO-based RPA attacks [16], we tested RO-based RPA attacks against *UCloD-Xilinx* implementations. We tested a different number of RO-sensors (32 RO-sensors, 64 RO-sensors, 96 RO-sensors, and 128 RO-sensors) to attack unprotected AES implementation. We found that a minimum of 128 RO-sensors is required to reveal the secret key within 20,000 encryptions. We tested unprotected AES implementation (used for other attacks in this paper) against 128 ROs. Our observations also aligns with [16] where authors stated that compared to a single TDC sensor a large number of RO-sensors are needed to collect sufficient voltage fluctuations to carry out a successful RPA attack. The CPA attack results for unprotected AES are shown in Figure 16. According to Figure 16, the secret key byte (key byte 0) can be revealed within 20,000 encryptions.

CPA attacks on *UCloD-Xilinx(N = 32)*, *UCloD-Xilinx(N = 64)* and *UCloD-Xilinx(N = 96)* revealed the secret key in less than 100,000 encryptions. *UCloD-Xilinx(N = 128)* revealed the secret key around 480,000 encryptions. *UCloD-Xilinx(N = 128)* provides ≈24× increased PA attack resistance compared to the unprotected AES implementation.

The CPA attacks results for *UCloD-Xilinx(N = 160)* is shown in Figure 17. As shown in the Figure 17, *UCloD-Xilinx(N = 160)* did not reveal the secret key for up to one million encryptions even when 128 RO sensors are used for the RPA attack.

B. PA ATTACKS USING OSCILLOSCOPE

We also carried out PA attacks (traditional power analysis attacks by measuring the power dissipation using an oscilloscope or ADC [13], [48]) using a Keysight DSOS404A oscilloscope to check the efficacy of eradicating PA attacks by *UCloD* methodology. Techniques were used that were similar to obtaining the RPA attack success rate of the secret key, on PA attacks. The CPA attack results for unprotected AES running on SASEBO GIII are shown in Figure 18. The secret key was revealed after 4,000 encryptions.

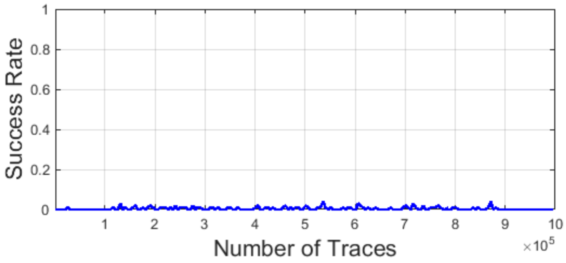


FIGURE 17. CPA attacks based on *UCloD-Xilinx(N = 160)* measured using 128 instances of RO-based power sensors.

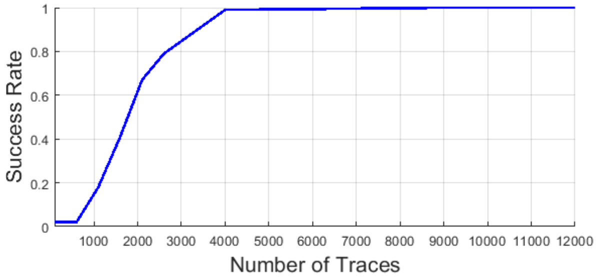


FIGURE 18. CPA attacks on using power dissipation (PA attack) from oscilloscope– unprotected AES implementation.

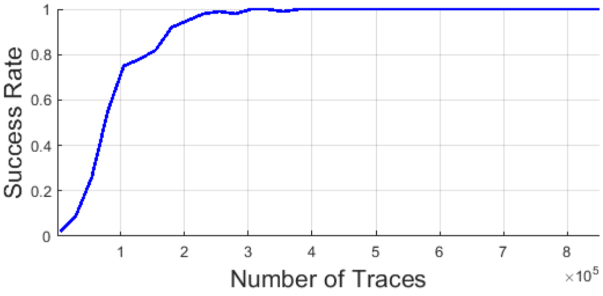


FIGURE 19. CPA attacks on *UCloD-Xilinx(N = 128)* using power dissipation measured from oscilloscope.

CPA attacks against *UCloD-Xilinx(N = 128)* implementation is presented in the Figure 19. The secret key byte 0 was revealed after 300,000 encryptions. *UCloD-Xilinx(N = 128)* provides ≈70× increased PA attack resistance compared to the unprotected AES implementation.

CPA attacks against *UCloD-Xilinx(N = 160)* implementation is presented in the Figure 20. The secret key was not revealed after one million encryptions, the SR reached 0.86 for one million encryptions. *UCloD-Xilinx(N = 160)* implementation provides greater than 250× increased PA attack resistance compared to the unprotected AES implementation.

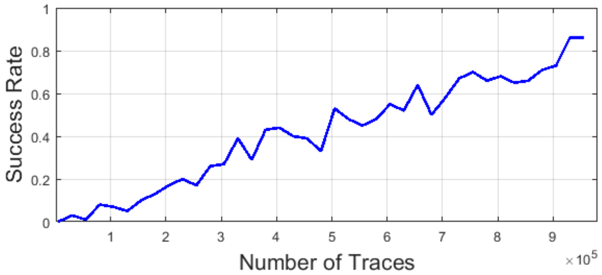


FIGURE 20. CPA attacks on *UCloD-Xilinx(N = 160)* using power dissipation measured from oscilloscope.

TABLE 5. State of the art PA attack countermeasures against *UcloD*.

	Countermeasure	Overhead		Attack
		Area	Time	Resistance
Balancing	WDDL [49]	5.00×[¶]	2×	-
	DPL [50]	4.50×	3.50×	-
	DAWDDL [51]	8.76×	-	-
	IWDDL [52]	4.34×	1.01×	-
	MDPL [53]	3.67×	1.43×	-
	QuadSeal [54]	6.50×	4.00×	$> 1M$
	NORA [55]	8.10×	-	$> 1M$
Masking	Trichina [56]	7.64×	-	-
	RSM [57]	1.41[*]×	-	$> 150K$
	Zeng et al. [58]	4.82×	1.46×	-
Randomizing	Lu et al. [59]	1.81×	1.63×	-
	Guneysu et al. [27]	-	3.77×	$3M$
	Fritzke et al. [32]	1.02×	-	$> 3M$[★]
	RFTC [25]	1.3×	1.7×	$> 3M$
	UCloD[†]	1.05×	1.01×	$> 1M$

— Not discussed/ disclosed in the manuscript

[¶] according to authors in [60]

[*] according to authors in [61]

[★] unprotected AES required 500K to reveal the key, thus PA attack resistance demonstrated is 6×

[†] *UCloD-Xilinx(N=160)* implementation

Table 5 shows comparison of the state of the art PA attack countermeasures against the *UCloD* implementation (we chose *UCloD-Xilinx(N = 128)* which is demonstrated CPA attack resistance for up to 300,00 encryptions). The first column of Table 5 depicts the countermeasure type, the second column depicts the state of the art PA countermeasures, the third column depicts area and time overheads, and the fourth column depicts the PA attack resistance demonstrated by each countermeasure. According to the Table 5, *UCloD-Xilinx(N = 160)* is the smallest PA attack countermeasure in both area and time overheads and shown secure up to one million encryptions (randomization methodology presented in [32] needed 500k traces to reveal the secret key byte of the unprotected encryptions where as our experimental setup needed only 4,000 encryptions as shown in Figure 18 to reveal the secret key byte of the unprotected encryption circuit).

VIII. DISCUSSION & FUTURE WORK

UCloD proposed in this paper shows increased RPA (TDC sensor-based and RO-sensor based attacks) and PA attack resistance when the size of the TDL is increased (N).

As the future work, we propose to implement *UCloD* on Xilinx Ultra Scale FPGAs where the tapped delay line has 512 taps and on Intel Altera FPGAs to test the effectiveness of mitigating RPA attacks. Adding additional and efficient TDC sensors to detect the random clock delays is also proposed as future work.

IX. CONCLUSION

FPGA security in the quest to confidentiality and security of the hardware designs running on FPGAs. RPA attacks impose unprecedented threats of being able to measure the voltage fluctuations occurring in the PDN of FPGAs. Such threats are becoming prominent with multi-tenant FPGA models where a single FPGA is shared among multiple users. The *UCloD*, clock delay methodology proposed in this paper, is shown to effectively mitigate RPA attack vulnerabilities and is demonstrated on a Xilinx FPGA and a Lattice Semiconductor FPGA. *UCloD* used TDLs, such as Xilinx IDELAY components, fabricated into FPGAs to generate clock delays. *UCloD* showed how small delays are inferred by TDC sensor-based delay sensors as readings, and how the inferred delays can be used to mitigate RPA attacks, as well as traditional PA attacks. *UCloD* demonstrated the scalability of increasing the range of delays induced in the clock and the RPA attack resistance which is necessary to create delays with large range when higher randomness is needed. *UCloD-Xilinx(N = 128)* and *UCloD-Lattice(N = 16)* implementations are the most robust countermeasures which have thus far been demonstrated against TDC sensor-based RPA attacks and *UCloD-Xilinx(N = 160)* is the most robust countermeasures which have thus far been demonstrated against RO sensor-based RPA attacks.*UCloD* must be tested against more powerful attacks, such as deep learning based side channel attacks, to test the efficacy of the randomness of the cryptographic clock to mitigate RPA attacks.

APPENDICES

We have added source code of TDL instantiation on Lattice FPGAs and Xilinx 7 Series FPGAs.

A. VERILOG CODE- LATTICE

```


SB_PLL40_2F_CORE
    lattice_pll2_inst(
        .REFERENCECLK(REFERENCECLK), // 1-bit  input: clock input
        .PLLOUTCOREA(PLLOUTCOREA),   // clock output to AES
        .PLLOUTCOREB(PLLOUTCOREB),  // not used
        .PLLOUTGLOBALA(PLLOUTGLOBALA),
        .PLLOUTGLOBALB(PLLOUTGLOBALB),
        .EXTFEEDBACK(),
        .DYNAMICDELAY(DELAY), // Random DELAY [0:3] from LFSR
        .RESETB(RESETN),
        .BYPASS(1'b0),
        .LATCHINPUTVALUE(),
        .LOCK(),
        .SDI(),
        .SDO(),
        .SCLK());

//\\ Fin=12, Fout=12;
defparam lattice_pll2_inst.DIVR = 4'b0000;
defparam lattice_pll2_inst.DIVF = 7'b0111111;
defparam lattice_pll2_inst.DIVQ = 3'b101;
defparam lattice_pll2_inst.FILTER_RANGE = 3'b001;
defparam lattice_pll2_inst.FEEDBACK_PATH = "SIMPLE";
defparam lattice_pll2_inst.DELAY_ADJUSTMENT_MODE_FEEDBACK = "DYNAMIC";
defparam lattice_pll2_inst.FDA_FEEDBACK = 4'b0000;
defparam lattice_pll2_inst.DELAY_ADJUSTMENT_MODE_RELATIVE = "FIXED";
defparam lattice_pll2_inst.FDA_RELATIVE = 4'b0000;
defparam lattice_pll2_inst.SHIFTREG_DIV_MODE = 2'b00;
defparam lattice_pll2_inst.PLLOUT_SELECT_PORTA = "GENCLK_HALF";
defparam lattice_pll2_inst.PLLOUT_SELECT_PORTB = "GENCLK_HALF";
defparam lattice_pll2_inst.ENABLE_ICEGATE_PORTA = 1'b0;
defparam lattice_pll2_inst.ENABLE_ICEGATE_PORTB = 1'b0;

endmodule
```

B. VERILOG CODE- XILINX 7 SERIES - FPGA

1) IDELAYE2 INSTANTIATION

```
(* IODELAY_GROUP = "Group_ADS1" *)    // group name should be
                                      //identical in idelay instances
IDELAYCTRL IDELAYCTRL_inst (
    .RDY(), // 1-bit output: Ready output
    .REFCLK(CLKREF), // 1-bit input: Reference clock input
    .RST(RESET) // 1-bit input: Active high reset input
);

(* IODELAY_GROUP = "Group_ADS1" *)
IDELAYE2 #(
    .CINVCTRL_SEL("FALSE"), // Enable dynamic clock inversion
    .DELAY_SRC("DATAIN"), // Delay input
    .HIGH_PERFORMANCE_MODE("TRUE"), // Reduced jitter ("TRUE"),
                                      //Reduced power ("FALSE")
    .IDELAY_TYPE("VAR_LOAD_PIPE"), // FIXED, VARIABLE, VAR_LOAD,
                                      //VAR_LOAD_PIPE
    .IDELAY_VALUE(0), // Input delay tap setting (0-31)
    .PIPE_SEL("TRUE"), // Select pipelined mode, FALSE, TRUE
    .REFCLK_FREQUENCY(192.0), // IDELAYCTRL clock input frequency
                                  //in MHz (190.0-210.0).
.SIGNAL_PATTERN("CLOCK") // DATA, CLOCK input signal
)
IDELAYE2_inst0 (
    .CNTVALUEOUT(), // 5-bit output: Counter value output
    .DATAOUT(CLK5), // 1-bit output: Delayed data output
                     // Clock signal to the
    .C(CLK), // 1-bit input: Clock input
    .CE(1'b0), // 1-bit input: Active high enable inc./dec.
    .CINVCTRL(1'b0), // 1-bit input: Dynamic clock inversion input
    .CNTVALUEIN(DELAY), // 5-bit input: Counter value input
                         // generated from LSFR
    .DATAIN(CLK5t), // 1-bit input: Internal delay data input
                     // Clock input from MMCM
    .IDATAIN(), // 1-bit input: Data input from the I/O
    .INC(1'b0), // 1-bit input: Increment /
                     //Decrement tap delay input
    .LD(1'b1), // 1-bit input: Load IDELAY_VALUE input
    .LDPIPEEN(1'b1), // 1-bit input: Enable Uclod
    .REGRST(RESET) // 1-bit input: Active-high reset
);
```

REFERENCES

[1] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107593.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO'99*, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 388–397.

[3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side—channel(s)," in *Cryptographic Hardware and Embedded Systems—CHES 2002*, B. S. Kaliski, K. Koç, and C. Paar, Eds. Berlin, Germany: Springer, 2003, pp. 29–45.

[4] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Advances in Cryptology—CRYPTO'96*. Berlin, Germany: Springer, 1996, pp. 104–113.

[5] D. J. Bernstein, "Cache-timing attacks on AES," Dept. Math., Statist., Comput. Sci., Univ. Illinois Chicago, Chicago, IL, USA, Tech. Rep. cachetiming-20050414, 2005.

[6] C. Shen *et al.*, "Cache side-channel attacks and countermeasures," in *Proc. 26th Asia South Pacific Design Automat. Conf. (ASPDAC)*, Tokyo, Japan, Jan. 2021.

[7] "Federal information processing standards publication 197 announcing the advanced encryption standard (AES)," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. FIPS 197, 2001. [Online]. Available: https://csrc.nist.gov/publications/detail/fips/197/final

[8] D. C. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Berlin, Germany: Springer-Verlag, 2003.

[9] S. B. Örs, F. Gürkaynak, E. Oswald, and B. Preneel, "Power-analysis attack on an ASIC AES implementation," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, vol. 2, Washington, DC, USA: IEEE Computer Society, 2004, p. 546.

[10] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards," in *Proc. USENIX Workshop Smartcard Technol. (WOST)*. Berkeley, CA, USA: USENIX Association, 1999, p. 17.

[11] C. Luo, Y. Fei, L. Zhang, A. A. Ding, P. Luo, S. Mukherjee, and D. Kaeli, "Power analysis attack of an AES GPU implementation," *J. Hardw. Syst. Secur.*, vol. 2, no. 1, pp. 69–82, Mar. 2018.

[12] S. B. Örs, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA—First experimental results," in *Cryptographic Hardware and Embedded Systems—CHES 2003*. Berlin, Germany: Springer, 2003, pp. 35–50.

[13] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Advances in Information Security). New York, NY, USA: Springer-Verlag, 2007.

[14] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on FPGAs," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 1111–1116.

[15] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs," in *Proc. FPGA*. New York, NY, USA: Association for Computing Machinery, 2013, pp. 101–104.

[16] J. Gravellier, J.-M. Dutertre, Y. Teglia, and P. Loubet-Moundi, "High-speed ring oscillator based sensors for remote side-channel attacks on FPGAs," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig)*, Dec. 2019, pp. 1–8.

[17] J. Zhang and G. Qu, "Recent attacks and defenses on FPGA-based systems," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 12, no. 3, pp. 1–24, Sep. 2019.

[18] J. Krautter, D. Gnad, and M. Tahoori, "CPAmap: On the complexity of secure FPGA virtualization, multi-tenancy, and physical design," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, pp. 121–146, Jun. 2020.

[19] A. Bag, S. Patranabis, D. B. Roy, and D. Mukhopadhyay, "Cryptographically secure multi-tenant provisioning of FPGAs," Feb. 2018, *arXiv:1802.04136*. [Online]. Available: https://arxiv.org/abs/1802.04136

[20] R. Elnaggar, R. Karri, and K. Chakrabarty, "Multi-tenant FPGA-based reconfigurable systems: Attacks and defenses," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 7–12.

[21] Z. István, G. Alonso, and A. Singla, "Providing multi-tenant services with FPGAs: Case study on a key-value store," in *Proc. 28th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2018, pp. 119–1195.

[22] Amazon. *Amazon EC2 F1 Instances*. Accessed: Apr. 28, 2021. [Online]. Available: https://aws.amazon.com/ec2/instance-types/f1/

[23] O. Glamocanin, L. Coulon, F. Regazzoni, and M. Stojilovic, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, p. 4.

[24] J. Krautter, D. R. E. Gnad, F. Schellenberg, A. Moradi, and M. B. Tahoori, "Active fences against voltage-based side channels in multi-tenant FPGAs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[25] D. Jayasinghe, A. Ignjatovic, and S. Parameswaran, "RFTC: Runtime frequency tuning countermeasure using FPGA dynamic reconfiguration to mitigate power analysis attacks," in *Proc. 56th ACM/IEEE Annu. Design Autom. Conf. (DAC)*, Jun. 2019, pp. 1–6.

[26] D. Jayasinghe, A. Ignjatovic, and S. Parameswaran, "SCRIP: Secure random clock execution on soft processor systems to mitigate power-based side channel attacks," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–7.

[27] T. Güneysu and A. Moradi, "Generic side-channel countermeasures for reconfigurable devices," in *Proc. CHES*. Berlin, Germany: Springer-Verlag, 2011, pp. 33–48.

[28] P. Ravi, S. Bhasin, J. Breier, and A. Chattopadhyay, "PPAP and iPPAP: PLL-based protection against physical attacks," in *Proc. ISVLSI*, Jul. 2018, pp. 620–625.

[29] J.-S. Coron and I. Kizhvatov, "Analysis and improvement of the random delay countermeasure of CHES 2009," in *Proc. CHES*. Berlin, Germany: Springer, 2010, pp. 95–109.

[30] Y. Lu, M. P. O'Neill, and J. V. McCanny, "FPGA implementation and analysis of random delay insertion countermeasure against DPA," in *Proc. Int. Conf. Field-Program. Technol.*, Dec. 2008, pp. 201–208.

[31] M. Bucci, R. Luzzi, M. Guglielmo, and A. Trifiletti, "A countermeasure against differential power analysis based on random delay insertion," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, May 2005, pp. 3547–3550.

[32] A. W. Fritzke, "Obfuscating against side-channel power analysis using hiding techniques for AES," M.S. thesis, Dept. Air Force, Air Univ., Islamabad, Pakistan, 2012.

[33] J. Tatsukawa. (2017). *MMCM and PLL Dynamic Reconfiguration*. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp888_7Series_DynamicRecon.pdf

[34] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems—CHES 2004*, M. Joye, and J.-J, Quisquater, Eds. Berlin, Germany: Springer, 2004, pp. 16–29.

[35] W. Kirch, Ed., *Pearson's Correlation Coefficient*. Dordrecht, The Netherlands: Springer, 2008, pp. 1090–1091.

[36] R. Siddiqui, F. Yuan, and Y. Zhou, ''A 500-MS/s 8.4-ps double-edge successive approximation TDC in 65 nm CMOS,'' in *Proc. IEEE 62nd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2019, pp. 770–773.

[37] Xilinx. (2015). *Cost-Optimized Portfolio Product Tables and Product Selection Guide*. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/ultrascale-plus-fpga-product-selection-guide.pdf

[38] X. Hu, L. Zhao, S. Liu, J. Wang, and Q. An, ''A stepped-up tree encoder for the 10-ps wave union TDC,'' in *Proc. 18th IEEE-NPSS Real Time Conf.*, Jun. 2012, pp. 1–5.

[39] S. M. Ismail, A. B. M. S. Rahman, and F. T. Islam, ''Low power design of Johnson counter using clock gating,'' in *Proc. 15th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2012, pp. 510–517.

[40] Xilinx. (2018). *7 Series FPGAs SelectIO Resources, User Guide*. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf

[41] Xilinx. (2016). *LogiCORE IP Utility IDELAYCTRL Logic (V1.0)*. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/util_idelay_ctrl/v1_0/pb044-util-idelay-ctrl.pdf

[42] Lattice Semiconductor. (Jun. 2016). *iCE40 sysCLOCK PLL Design and Usage Guide*. [Online]. Available: https://www.latticesemi.com/-/media/LatticeSemi/Documents/ApplicationNotes/IK/iCE40sysCLOCKPLLDesignandUsageGuide.ashx?document_id=47778

[43] Y. Hori, T. Katashita, A. Sasaki, and A. Satoh, ''SASEBO-GIII: A hardware security evaluation board equipped with a 28-nm FPGA,'' in *Proc. 1st IEEE Global Conf. Consum. Electron. (GCCE)*, Oct. 2012, pp. 657–660.

[44] J. Krautter, D. R. E. Gnad, F. Schellenberg, A. Moradi, and M. B. Tahoori, ''Software-based fault and power side-channel attacks inside multi-tenant FPGAs,'' in *Proc. Demo Session, IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Tysons, VA, USA, May 2019. [Online]. Available: http://www.hostsymposium.org/host2019/host_2019awards.php

[45] A. Thillard, E. Prouff, and T. Roche, ''Success through confidence: Evaluating the effectiveness of a side-channel attack,'' in *Cryptographic Hardware and Embedded Systems—CHES 2013*, G. Bertoni and J.-S. Coron, Eds. Berlin, Germany: Springer, 2013, pp. 21–36.

[46] D. Fledel and A. Wool, ''Sliding-window correlation attacks against encryption devices with an unstable clock,'' in *Proc. 25th Int. Conf. Sel. Areas Cryptogr.*, Calgary, AB, Canada, Aug. 2018, pp. 193–215.

[47] L. Batina and J. Hogenboom, ''Principal component analysis and side-channel attacks,'' M.S. thesis, Dept. Comput. Sci., Radboud Univ., Nijmegen, The Netherlands, 2010, pp. 536–539.

[48] D. Jayasinghe, R. Ragel, J. A. Ambrose, A. Ignjatovic, and S. Parameswaran, ''Advanced modes in AES: Are they safe from power analysis based side channel attacks?'' in *Proc. IEEE 32nd Int. Conf. Comput. Design (ICCD)*, Oct. 2014, pp. 173–180.

[49] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, ''Prototype IC with WDDL and differential routing—DPA resistance assessment,'' in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Berlin, Germany, 2005, pp. 354–365.

[50] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev, ''Design and analysis of dual-rail circuits for security applications,'' *IEEE Trans. Comput.*, vol. 54, no. 4, pp. 449–460, Apr. 2005.

[51] A. Wild, A. Moradi, and T. Güneysu, ''Evaluating the duplication of dual-rail precharge logics on FPGAs,'' in *Constructive Side-Channel Analysis and Secure Design*. Cham, Switzerland: Springer, 2015, pp. 81–94.

[52] R. P. McEvoy, C. C. Murphy, W. P. Marnane, and M. Tunstall, ''Isolated WDDL: A hiding countermeasure for differential power analysis on FPGAs,'' *ACM Trans. Reconfigurable Technol. Syst.*, vol. 2, no. 1, pp. 1–23, Mar. 2009.

[53] T. Popp and S. Mangard, ''Masked dual-rail pre-charge logic: DPA-resistance without routing constraints,'' in *Proc. CHES*, vol. 3659, J. R. Rao and B. Sunar, Eds., 2005, pp. 172–186.

[54] D. Jayasinghe, A. Ignjatovic, J. A. Ambrose, R. Ragel, and S. Parameswaran, ''QuadSeal: Quadruple algorithmic symmetrizing countermeasure against power based side-channel attacks,'' in *Proc. Int. Conf. Compil., Archit. Synth. Embedded Syst. (CASES)*, Oct. 2015, pp. 21–30.

[55] D. Jayasinghe, A. Ignjatovic, and S. Parameswaran, ''NORA: Algorithmic balancing without pre-charge to thwart power analysis attacks,'' in *Proc. 30th Int. Conf. VLSI Design 16th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2017, pp. 167–172.

[56] E. Trichina, ''Combinational logic design for AES subbyte transformation on masked data,'' *IACR Cryptol. ePrint Arch.*, vol. 2003, p. 236, Nov. 2003.

[57] M. Nassar, Y. Souissi, S. Guilley, and J. Danger, ''RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs,'' in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2012, pp. 1173–1178.

[58] J. Zeng, Y. Wang, C. Xu, and R. Li, ''Improvement on masked S-box hardware implementation,'' in *Proc. Int. Conf. Innov. Inf. Technol. (IIT)*, Mar. 2012, pp. 113–116.

[59] Y. Lu, M. O'Neill, and J. McCanny, ''Evaluation of random delay insertion against DPA on FPGAs,'' *ACM Trans. Reconfigurable Technol. Syst.*, vol. 4, no. 1, pp. 11:1–11:20, Dec. 2010.

[60] N. Selmane, S. Bhasin, S. Guilley, T. Graba, and J.-L. Danger, ''WDDL is protected against setup time violation attacks,'' in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, Sep. 2009, pp. 73–83.

[61] H. Huang, L. Liu, Q. Huang, Y. Chen, S. Yin, and S. Wei, ''Low area-overhead low-entropy masking scheme (LEMS) against correlation power analysis attack,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 2, pp. 208–219, Feb. 2019.

DARSHANA JAYASINGHE received the B.S. degree in computer engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2010, and the Ph.D. degree in computer science and engineering from the University of New South Wales, in 2017. He is currently working as a Research Associate with the University of New South Wales. His research interests include side channel analysis attacks, side channel analysis attack countermeasures, and hardware security.

ALEKSANDAR IGNJATOVIC received the Ph.D. degree in mathematical logic from the University of California, Berkeley. He is currently an Associate Professor with the School of Computer Science and Engineering, University of New South Wales, Australia. His current research interests include approximation theory, sampling theory and applied harmonic analysis, embedded systems, and algorithm design.

SRI PARAMESWARAN received the B.Eng. degree from Monash University and the Ph.D. degree from The University of Queensland. He is currently a Professor with the School of Computer Science and Engineering, University of New South Wales. His research interests include system level synthesis, low-power systems, high-level systems, and network on chips. He has served on the Program Committees for Design Automation Conference (DAC), Design and Test in Europe (DATE), the International Conference on Computer Aided Design (ICCAD), the International Conference on Hardware/Software Code-Sign and System Synthesis (CODES-ISSS), and the International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES). He served as the Editor-in-Chief for the IEEE Embedded Systems Letters, from 2016 to 2019. He serves or has served on the editorial boards for IEEE Transactions on Computer Aided Design, *ACM Transactions on Embedded Computing Systems*, the *EURASIP Journal on Embedded Systems*, and the *Design Automation for Embedded Systems*.