IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 27, 2021, accepted November 5, 2021, date of publication November 9, 2021, date of current version November 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126703

Correlation Power Analysis Attack Resisted Cryptographic RISC-V SoC With Random Dynamic Frequency Scaling Countermeasure

BA-ANH DAO[1,2], (Graduate Student Member, IEEE), TRONG-THUC HOANG[1,3], (Graduate Student Member, IEEE), ANH-TIEN LE[1,2], (Graduate Student Member, IEEE), AKIRA TSUKAMOTO[3], KUNIYASU SUZAKI[3,4], (Member, IEEE), AND CONG-KHA PHAM[1], (Member, IEEE)

[1]Department of Computer and Network Engineering, The University of Electro-Communications (UEC), Chofu, Tokyo 182-8585, Japan
[2]Academy of Cryptography Techniques (ACT), Hanoi 12511, Vietnam
[3]National Institute of Advanced Industrial Science and Technology (AIST), Tokyo 135-0064, Japan
[4]Technology Research Association of Secure IoT Edge Application Based on RISC-V Open Architecture (TRASIO), Tokyo 101-0022, Japan

Corresponding author: Ba-Anh Dao (daobaanh@vlsilab.ee.uec.ac.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP16007.

ABSTRACT Cryptographic System-on-Chips (SoCs) are becoming more and more popular. In these systems, cryptographic accelerators are integrated with processor cores to provide users with the software's flexibility and hardware's high performance. First, this work aimed to confirm the vulnerability of cryptographic SoCs against several types of power analysis attacks. Then, the novel Random Dynamic Frequency Scaling (RDFS) countermeasure is proposed to improve the resistance of such systems. The proposed RDFS countermeasure improved the power analysis resistance while maintaining low-performance overhead and hardware costs by generating more than 219,000 distinct frequencies for driving only the cryptographic accelerators. The effectiveness of the proposed RDFS countermeasure is demonstrated by conducting realistic Correlation Power Analysis (CPA) attacks, Deep-Learning-based Side-Channel Analysis (DL-SCA) attacks, and Test Vector Leakage Assessment (TVLA) testing methodology. The experimental results show that the RISC-V SoC protected by RDFS countermeasure can withstand CPA attacks and TVLA test with more than five million power traces, which is the best result compared to other related works. The proposed RDFS countermeasure also harden the targeted RISC-V SoC's resistance against DL-SCA attacks.

INDEX TERMS Side-channel attacks, correlation power analysis attacks, deep learning based side-channel attacks, countermeasure, RISC-V.

I. INTRODUCTION

Nowadays, utilizing heterogeneous System-on-Chips (SoC) is the prominent solution to realize high-performance embedded systems. Advances in VLSI technologies allow integrating all or most components of a conventional computer system into a single SoC. These integrated components almost always include at least one to several processor cores, various memory blocks and multiple peripheral circuits or hardware accelerators. The processor cores can be a microcontroller, a microprocessor or an application-specific processor. It executes the software programs stored in memory space. Users are free to reprogram, modify and improve the software to adapt to any general tasks. On the other hand, the peripheral circuits and hardware accelerators provide more advanced performance for some heavy, specific tasks. Examples of these tasks could be tensor processing, cryptographic computation, data collection. Therefore, the SoC architecture offers versatile and reliable solutions for a wide range of applications since it combines programmable software's flexibility and customized hardware accelerators' performance.

The RISC-V open instruction set architecture (ISA) is recently published under open-source licenses by the

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

RISC-V Foundation. It has become an exciting topic for numerous SoC designers from academia and industry since it removes most of the limitations of working with proprietary designing tools and SoC's components. Designers can easily create their own RISC-V SoC that is suitable to their unique demands. In just a few years, RISC-V SoCs have become very popular and are applied in many fields. For instance, Feng *et al.* integrated the NVIDIA Deep Learning Accelerator (NVDLA) into a RISC-V SoC to run the LeNet-5 and accelerate the handwritten numeral recognition process up to 4,647 times [1]. Zhong *et al.* present a RISC-V SoC with an integrated visible light communication (VLC) module for mobile payment applications [2]. A low power, medical implantable RISC-V SoC for tissue stimulation is proposed by Arnaud *et al.* [3]. Among all these various applications, security is increasingly attracting attention. Many researchers and designers realized that the openness and flexibility of the RISC-V SoC could be used to implement their ideas for improving the performance and reliability of a secure embedded system. Consequently, cryptographic accelerators are integrated into RISC-V SoCs for different purposes. For example, in [4], Banerjee *et al.* introduce a RISC-V SoC equipped with an energy-efficient reconfigurable cryptographic engine, which includes hardware accelerators for computation of Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), and Secure Hash Algorithm (SHA). They proposed to apply their design to Datagram Transport Layer Security (DTLS) applications. A RISC-V core integrated with an AES-128 encryption engine is designed for IoT applications by Zang *et al.* [5]. In [6], Duran *et al.* accelerated AES-256 by using a RISC-V SoC with custom instructions and an enhanced memory access scheme.

In 1999, the Power Analysis attack was first introduced by Kocher *et al.* [7]. It is categorized as a type of side-channel attack since the attack does not target the mathematical weaknesses of the cryptographic algorithms but instead exploits the unintended information leakage caused by their implementation. The leaked information could be obtained by gathering and analyzing the power consumption or the electromagnetic radiation of the targeted device when it performs cryptographic functions. Power analysis attacks have rapidly grown over the last two decades and have become a powerful and popular method of breaching the security of cryptographic systems. Various works demonstrated that power analysis attacks could easily compromise the security of different cryptographic implementations, such as software implementations on general-purpose microcontrollers [8] or smart-cards [9], standalone hardware implementations in field programmable gate arrays (FPGAs) [10] or in application-specific integrated circuits (ASICs) [11]. In these implementations, the majority of the circuit's components are involved in the cryptographic operation. Therefore, the power traces measured from these devices contain a significant amount of leaked information and only a small amount of noise, favouring the adversaries. In contrast, attacking complex devices such as SoC with integrated cryptographic accelerators is challenging since the cryptographic circuits are packaged with other SoC's components (such as processing cores, bus interconnect system, other peripherals, accelerators), and they are typically supplied from the same off-chip power source. The exploitable power consumption caused by cryptographic accelerators cannot be isolated from the significant, unexploitable power consumption of other SoC's components. In other words, the power traces obtained from the cryptographic SoC have a high level of noise or even be heavily distorted, which cause certain difficulties for attackers.

Consequently, even though various cryptographic SoC has been proposed, there is only a reduced number of works where the system's resistance against Power Analysis attack is concerned. For instance, some occasional works that can be found are [12], [13]. In [12], Cai *et al.* perform experiments on an SoC with a cryptographic coprocessor running the 128-bit AES algorithm [12]. The total power consumption traces of the SoC, including the power consumption of the CPU and the AES-128 coprocessor, are acquired and analyzed to reveal the secret key used in AES-128 encryption. In [13], Hettwer *et al.* reported attacks results on an AES core implemented on the Programmable Logic part of a Xilinx Zynq Ultrascale+ SoC. However, according to Xilinx's Zynq SoC design, the Programmable Logic part is supplied via a separated power pin [14]. Hence, Hettwer *et al.* can acquire and analyze the electromagnetic traces emitted from the decoupling capacitor of the Programmable Logic's power supply, which solely contains the exploitable information.

Due to the lack of publications where authors discuss the vulnerability of overall cryptographic SoC and conduct proper security evaluation against Power Analysis attacks, there are even fewer works that provide concrete results on countering and preventing Power Analysis attacks for cryptographic SoC. Therefore, in this work, we first provide a demonstration to confirm that realistic Power Analysis attacks on integrated cryptographic SoC are feasible. Then, we propose a design method to enhance the resistance against the Power Analysis attack of the integrated cryptographic SoC. The proposed method is named Random Dynamic Frequency Scaling (RDFS) since its concept is to randomly alter the clock frequency of only the cryptographic accelerators after each encryption/decryption. We demonstrate the effectiveness of the proposed method by applying it to a RISC-V SoC with AES-128 hardware accelerator, implementing the whole RISC-V SoC on the Sakura-X FPGA board, and evaluating its security against Power Analysis attacks. The security evaluation is done by conducting the side-channel leakage test called Test Vector Leakage Assessment (TVLA) and performing CPA attacks. Furthermore, we also test the designed RISC-V SoC with the state-of-the-art profiled Deep Leaning Side-Channel Analysis (DL-SCA) attacks. In our example RISC-V SoC design, the proposed method allows the AES-128 accelerator to operate with more than 219,000 different frequencies. These operating frequencies of the

AES-128 accelerator are randomly selected by an on-chip Pseudo-Random Number Generator (PRNG). The selected frequency values are random and uniformly distributed in the range of 50 MHz to 100 MHz. The lower limit of 50 MHz is the maximum operating frequency of the whole RISC-V SoC without applying the RDFS countermeasure, while the upper limit of 100 MHz is the maximum operating frequency of the standalone AES-128 accelerator.

A. RELATED WORKS

Power Analysis attacks are classified into two groups based on their approach and applied scenarios. These two groups are non-profiled power analysis attacks and profiled power analysis attacks. In non-profiled power analysis attacks, the attackers only have access to the targeted device and gather its physical leakage. They also have minimal knowledge about the power consumption characteristics of the targeted device and can only construct hypothetical power models to use in the statistical analysis process. Some popular examples of non-profiled power analysis attacks are the Simple Power Analysis (SPA) attack, the Differential Power Analysis (DPA) attack, and the Correlation Power Analysis (CPA) attack. The SPA attack and the DPA attack are presented by Kocher *et al.* [7]. In the SPA attack, a power trace or an electromagnetic trace measured during a device's operation is visually inspected to obtain information about the timing and type of the processed cryptographic operation. In the DPA attack, multiple power traces or electromagnetic traces are statistically analyzed to derive the most likely key block used in related cryptographic operations. In 2004, Brier *et al.* proposed using Pearson's correlation coefficient as a statistical parameter to improve the efficiency of the power analysis attacks [15]. They named their method as CPA attack. Another set of power analysis attacks is the profiled power analysis attack, which can only be applied if the attackers possess an open, identical copy of the targeted device. The attackers use that copy to characterize the physical leakage and construct a better power model that perfectly fits the targeted device's actual power consumption characteristics. Hence, the profiled power analysis attacks could outperform the non-profiled power analysis attacks in terms of the number of required attacking power traces. However, having an identical copy of the targeted device is impractical, especially when the targeted devices are flexible and highly customizable, such as RISC-V SoCs. Some examples of profiled power analysis attacks are Template Attacks [16], and Linear Regression Analysis [17]. Machine Learning (ML) and Deep Learning (DL) techniques are also utilized in profiled power analysis attacks because of the strong similarity between supervised learning and profiled attacks. For instances, several ML-based profiled power analysis attacks are presented by Chakraborty *et al.* [18], by Duan *et al.* [19], and by Hou *et al.* [20]. Some DL-based profiled power analysis attacks were presented by Maghrebi *et al.* [21], by Cagli *et al.* [22], and by Benadjila *et al.* [23]. Furthermore, in 2019. Timon *et al.* demonstrated that DL-based attacks could also be applied in the non-profiled scenarios [24].

Power Analysis attacks can reveal the secret key due to the dependence of devices' instantaneous power consumption on processing intermediate values. These intermediate values are derived by functions of a subkey and a known non-constant data, where the subkey is a small part of the secret key, and the known non-constant data is either part of plaintext input or ciphertext output. Various works on Power Analysis countermeasures have been published. Their goal is to break the link between the cryptographic device's power consumption and the intermediate values. Based on their approaches, all published countermeasures could be classified into two major groups. The first group can be named as hiding countermeasures. These countermeasures try to modify the power consumption characteristics of the cryptographic device so that its data dependency of instantaneous power is reduced significantly. For instance, some popular hiding countermeasures are reducing exploitable signal to noise ratio [25], randomly changing supply voltage, and clock frequency [26], maintaining approximately the same level of power consumption by adding filters [27], using current flattening circuit [28], or implementing on DPA-resistant logic styles [29]–[31]. Hiding countermeasures do not modify the cryptographic algorithms. The devices protected by hiding countermeasures still process the same intermediate values as the unprotected devices. This is the main difference between hiding countermeasures and the other countermeasures group called masking. Devices that are protected by masking countermeasures will process randomized intermediate values instead of conventional values. Thus, the cryptographic algorithm needs to be modified to adapt to the utilization of random masking. This modification could significantly increase the cryptographic algorithm's computational complexity and reduce the corresponding implementations' throughput and performance while increasing hardware resource utilization. Several examples of masking countermeasures are Boolean masking, arithmetic masking [32], [33], secret sharing [34], [35], threshold cryptography [36], [37]. Aside from two major groups of countermeasures, a limited number of works are proposed to update the secret key after a certain time interval or number of encryption/decryption usages. They include the Fresh re-keying countermeasure by Medwed *et al.* [38], the Moving target defense mechanism by Vuppala *et al.* [39], and the Key Update countermeasure by Gui *et al.* [40]. The values of the updated secret key could be generated by software functions or embedded hardware structures such as strong Physical Unclonable Functions (PUFs) [41]. These key update countermeasures are demonstrated to be effective. However, they still cause significant overheads compared to fixed key usage and introduce additional challenges in managing and distributing the updated secret keys.

Each particular countermeasure has its advantages and disadvantages. In general, hiding countermeasures are more generic than masking countermeasures. When applying

masking countermeasures, each cryptographic implementation needs to adapt to a specific masking scheme. If multiple cryptographic algorithms are integrated into the same protected device, the overheads of multiple masked schemes are combined. Meanwhile, hiding countermeasures are typically deployed at the physical level of a device and cover multiple integrated cryptographic algorithms at once. Hence, the overheads caused by hiding countermeasures are often smaller than those of masking countermeasures.

Previous works proposed various effective hiding countermeasures, which share the general idea of randomizing the operating conditions of the targeted device to mitigate power analysis attacks [26], [42], [43]. Yang *et al.* suggested using Random Dynamic Voltage and Frequency Scaling (RDVFS) as a DPA countermeasure. Their idea is to alter the pair of cryptographic circuit's supply voltage-operating frequency randomly [26]. Later on, Baddam *et al.* proposed varying only the supply voltage instead of the voltage-frequency pair since they reported that the change in operating frequency (of a standalone cryptographic circuit) is easy to detect by visually observing the peaks in each power consumption trace [42]. In our previous work [43], the Random Dynamic Back-gate Bias (RDBB) is also proposed to enhance DPA resistance of devices fabricated in Fully Depleted Silicon-on-Insulator (FD-SOI) technology. However, these countermeasures can only be applied for cryptographic implementations on ASIC, not FPGA, since they require special features of fabricating technologies such as controlling the back-gate bias or operating in a wide range of supply voltage. Recently, there are several works reported to use randomizing only the clock signal of the cryptographic circuit as Power Analysis countermeasures [8], [10], [13], [44]. They have a similar approach, in which the on-chip circuits generate a large number of alternated clock signals from a constant off-chip oscillating source for driving cryptographic circuits. Unlike other randomizing countermeasures, the randomizing-clock-frequency-based countermeasure does not require any particular characteristic of the fabricating technology. Therefore, it can be implemented on ASICs as well as FPGAs.

Clock Managers are frequently used in previous works to generate different output clocks, which are subsequently driven to the cryptography circuit using clock multiplexers. In [44], Guneysu and Moradi use two Digital Clock Managers that are available in Xilinx FPGA to generate eight phase-shifted output clocks with the same clock frequency. A network of multiplexers, controlled by random selecting signals, combines all eight generated clock signals to produce a single clock output for the cryptographic core. This method effectively improves DPA resistance, but the combined clock signal has a very low pulse rate, leading to a significant cryptographic core's throughput downgrade. In [10], Jayasinghe *et al.* use the Mixed Mode Clock Managers (MMCM) primitive in Xilinx FPGA to generate up to 3,072 distinct frequencies in the range of 24 MHz to 48 MHz. All 3,072 distinct frequencies are divided into three sets. Each set of frequencies is outputted via an output clock port of the MMCM primitive. Then one of these three clock signals is randomly selected to drive each clock cycle of the cryptographic core. The authors reported that by carefully choosing the combination of distinct frequencies for each encryption, their protected AES-128 core could have 67,684 different encrypting completion times, varying between 208.33 *ns* and 833.32 *ns*. They considered that the higher number of different completion times is, the more significant misalignment in the measured power traces will be. Hence, the power analysis resistance improvement is achieved. Indeed, their experimental results showed that CPA attacks could not break the protected cryptographic core even with four million analyzed traces. Also, there is no detectable leakage in the TVLA test results with one million traces. The authors named this countermeasure as Runtime Frequency Tuning Countermeasure (RFTC). Later on, in [8], Jayasinghe *et al.* also apply a similar countermeasure to RISC-V, an open-source RISC-V Processor. They named this countermeasure SCRIP. The SCRIP LowRISC executes a software implementation of AES-128. It is demonstrated to be secured against CPA attacks with 300,000 analyzed traces and shows no first-order leakage in the TVLA test with 200,000 traces. In [13], Hettwer *et al.* presented a similar countermeasure named Dynamic Frequency Randomization (DFR). This countermeasure also exploits the on-the-fly ability of clock manager IPs on FPGAs to create approximately 2,000 distinct frequencies. However, their proposed method can only be applied to advanced SoCs, including a Processing System (PS) and a Programmable Logic (PL- also known as the FPGA part). The PS core generates a scalable input clock signal for the PL. Then, the PL part generates a highly randomized output clock signal for driving the cryptographic core in the similar manners as described in [8], [10]. Hettwer *et al.* reported that they could achieve more than 20 million different encrypting completion times for the AES-128 core. Their experimental results on a Xilinx Zynq UltraScale+ FPGA demonstrated that their proposed countermeasure could not be broken by CPA within one million power traces. It also passed the TVLA test with five million traces and was able to withstand powerful DL-SCA attacks.

In these previous papers, a different clock frequency is randomly chosen to drive the cryptographic circuit after each clock cycle. The corresponded authors consider the number of distinct encrypting completion times as a critical designing parameter, where the encrypting completion time is the sum of several clock cycles that a cryptographic circuit requires to execute an encrypting process. Their works try to achieve as many distinct completion times as possible. In our opinion, choosing encrypting completion times as a designing parameter is inappropriate due to the following reasons. First, in Power Analysis attacks, the attackers are only interested in the instantaneous power consumption of the targeted devices at the exact moments when the sensitive intermediate values are being executed. If these moments of interest are

well-aligned among the set of acquired power traces, the power analysis attacks are more likely to succeed. For example, we can consider a hardware AES-128 implementation that could execute each encryption in 10 clock cycles. The intermediate values that could be used when attacking an AES-128 implementation are the S-Box substitution outputs in the first round or the inputs of S-Box substitution in the last round [15]. In these cases, the moments of interest (*i.e.* points-of-interest or POIs) would be located in the first cycle or the last tenth cycle of the power trace, respectively. Second, different completion times do not guarantee the POIs' misalignment in the set of acquired power traces. For example, when using the S-Box substitution outputs in the first round as the intermediate values, POIs could still be aligned while the completion times are different if the first clock cycles of each trace are the same and the rest of the clock cycles are different. Therefore, our proposed RDFS countermeasure aims to generate as many distinct frequencies as possible, and the cryptographic circuit is kept at a fixed frequency during each encryption/decryption. Only then we can ensure that the POIs in the set of measured power traces are heavily misaligned.

B. CONTRIBUTIONS

The contributions of this paper are highlighted as follows.

1) This work provides practical experiments to confirm that attackers can successfully perform Power Analysis attacks on complex cryptographic SoC devices, where the measured power traces are extremely noisy. The exploitable power caused by the cryptographic hardware accelerator are comparable with or even overshadowed by the unexploitable power caused by other SoC's components.
2) A novel SoC designing technique is proposed as a countermeasure against power analysis attacks, which take the best out of the ability to dynamically reconfigure the operating frequency of specific cryptographic accelerators within the designated SoC. Compared with previous related works, the proposed technique can achieve the highest number of distinct frequencies.
3) The proposed technique's effectiveness in improving resistance against power analysis attacks is evaluated with different methods, including the TVLA leakage test, CPA attacks, and profiled DL-SCA attacks.

The rest of the paper is organized as follows: Section II discusses several necessary concepts for understanding the remainder of the paper. Our proposed SoC designing technique for improving DPA resistance is presented in Section III. Section IV provides the evaluation of the proposed countermeasure's effectiveness by different methods, where the experimental setup and results are described in detail. Section V discusses some limitations of this work and suggests future developments. Finally, Section VI concludes the paper.

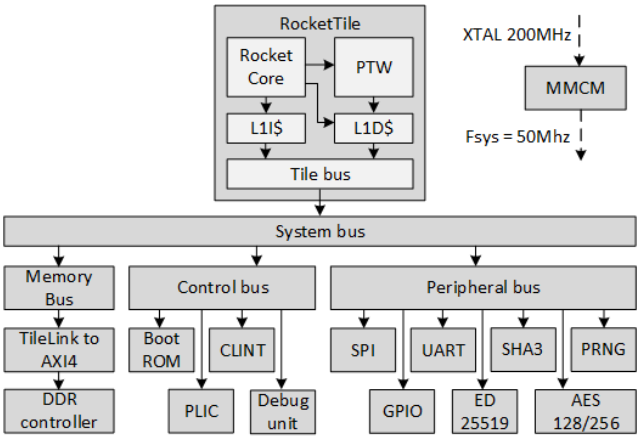

FIGURE 1. The targeted RISC-V SoC architecture.

II. PRELIMINARIES

A. TARGETED DEVICE

A 32-bit RISC-V SoC is developed and used as a targeted device in later experiments. This RISC-V SoC is a variant of the secure RISC-V system in our previous work [45], that features several cryptographic algorithm hardware accelerators to speed up the boot procedure of the Keystone Trust Exclusive Environment (TEE). The targeted SoC is generated by using the Chipyard framework [46]. The Chipyard framework provides various Chisel-based hardware generators and utilities for designers to use and develop custom RISC-V SoC. The architecture of the targeted SoC, developed in this work, is described in Figure 1. The SoC includes a 32-bit RISC-V Rocket core, which implements the RV32IMAC instruction set [47]. In other words, the integrated RISC-V core is compatible with **I**nteger, **M**ultiplication, **A**tomic, and **C**ompress extensions. The Rocket core utilizes a 16 KB instruction cache and a 16 KB Layer-1 data cache. The core complex is connected to an interconnected bus network, which consists of the System bus, Memory bus, Control bus, and Peripheral bus, using the TileLink protocol [48]. Similarly, each bus is used to attach related components. The DDR memory controller is attached to the Memory bus and supports up to 1 GB of Random-Access Memory (RAM), which provides execution memory space for the Linux operating system (OS). The Control bus is attached to several standard peripherals such as the BootROM, the Core-Local Interrupts (CLINT), the Platform-level Interrupt Controller (PLIC), and the Debug unit. The targeted SoC uses a similar boot procedure with the Freedom U540-C000's bootloader [49]. The zero stage bootloader (ZSBL), the first set of instructions to be executed when the SoC starts or comes out of reset, is stored in the 16 KB BootROM. The CLINT manages software interrupts and timer interrupts for the Rocket core. The PLIC combines and masks device interrupts and external interrupts. The Debug unit is used to control the SoC externally through standard JTAG protocol. Data and instructions can be written to or read from the system's memory

using the Debug unit. Meanwhile, the Peripheral bus is used to attach other memory-mapped input/output (MMIO) peripherals. They are controllers for Serial Peripheral Interface (SPI), General-purpose input/output (GPIO), Universal Asynchronous Receiver-Transmitter (UART), and some other cryptographic accelerators like the Advanced Encryption Standard (AES128/256), the Secure Hash Algorithm (SHA3), the Edwards-curve Digital Signature Algorithm (Ed25519), and the Pseudo-Random Number Generator (PRNG). The SPI controller drives an external SD-card, where the first stage bootloader and the Linux bootloader are located. The UART allows the targeted SoC to communicate with other devices via the UART communication protocol. The SHA3 and Ed25519 accelerators are used to speed up the boot procedure in our previous work [45]. In this paper, they served as noise sources to highlight the vulnerability of the AES128/256 accelerator even when it is integrated into complex, unprotected SoC devices. The AES128/256 accelerator is an open-source RTL design, which is available on Github [50]. According to the information on Github's repository [50], if implemented on an FPGA, this standalone AES accelerator can operate at a maximum clock frequency of around 100 MHz.

As mentioned earlier, the targeted SoC has a boot procedure similar to the Freedom U540-C000's bootloader. Therefore, it can operate in bare-metal mode like a simple microcontroller and operate with the Linux operating system like a microprocessor. The AES accelerator can be accessed via bare-metal programs or via programs overlaid on the Linux OS. When the SoC is in the bare-metal mode, the Rocket core only processes a waiting loop while the AES accelerator executes the encryption/decryption. Hence, the noise contributed by the Rocket core is repetitive. Meanwhile, when the SoC is in the operating system mode, other OS's processes can also be executed during the AES accelerator's encryption/decryption. In this case, the noise contributed by the Rocket core is more likely to be unpredictable. Therefore, power analysis attacks on the targeted SoC while operating in OS mode are more difficult to conduct than bare-metal mode.

B. CORRELATION POWER ANALYSIS ATTACK

Correlation Power Analysis attack was first introduced in 2004 by Brier *et al.* [15], which utilizes the Hamming Distance power model and the Pearson's Correlation Coefficient equation. CPA attacks also use the general attack strategy that is used by all DPA attacks. In this work, we perform CPA attacks on both protected and unprotected designs to evaluate the effectiveness of the proposed designing method. The practical attack strategy used in this work consists of the following steps. It is important to note that the CPA attack only targets one byte at a time. Therefore, at least 16 attacks are necessary to recover all 16-bytes of the secret key used in the AES-128 algorithm.

- **Step 1:** Choosing the intermediate value of the cryptographic algorithm that is processed by the targeted device. There are several options for choosing the intermediate value when attacking an AES implementation. We choose to use the S-Box substitution outputs in the first round as the intermediate value in this work.
- **Step 2:** Acquiring the power consumption traces of the targeted device while it is encrypting random plaintext inputs. For each encryption, the corresponding power trace should cover the whole encrypting interval. The length of a power trace is denoted as $\boldsymbol{T}$. The random plaintext input corresponding to each encryption also needs to be recorded. The number of measured traces is denoted as $\boldsymbol{D}$. Hence, after this step, we will obtain $\boldsymbol{D}$ plaintext and $\boldsymbol{D}$ corresponding power traces. The traces can also be represented as power trace data matrix of size $\boldsymbol{D}\times\boldsymbol{T}$.
- **Step 3:** Computing the hypothetical intermediate values matrix. By using (1), we calculate the hypothetical intermediate value matrix $I_{d,i}$ from D_d and K_i. D_d is the n^{th} byte of the d^{th} plaintext input, while K_i is the i^{th} key hypothesis, with $K_i = i-1$; $i \in (1, 256)$. In other words, we guess the value of K_i. Since the size of K_i is 01 byte, K_i could have a value in the range from 0 to 255. With each possible value of K_i and a known value of D_d, we calculated a possible value of S-box output in the 1st round.

$$I_{d,i} = Sbox(D_d \oplus K_i) \tag{1}$$

  The result of this calculation is a hypothetical intermediate value matrix of size $\boldsymbol{D}\times\boldsymbol{K}$, where $\boldsymbol{D}$ is the number of measured traces and $\boldsymbol{K}$ is the number of subkey hypotheses.
- **Step 4:** Computing hypothetical power consumption values matrix. Each value in the hypothetical intermediate value matrix is mapped to a hypothetical power consumption value by applying a suitable power consumption model. After mapping, we obtain the hypothetical power consumption matrix of size $\boldsymbol{D}\times\boldsymbol{K}$. In this work, based on experimental results, we found that using Hamming Weight models would provide better attack results. Then, the hypothetical power consumption value is computed by (2).

$$H_{d,i} = HW(I_{d,i}) \tag{2}$$

  where:
  $HW()$ is the Hamming weight function, which counts the number of bits "1" in the binary input.
  $I_{d,i}$ is the hypothetical intermediate value matrix obtained in step 3.
  $H_{d,i}$ is the hypothetical power consumption value matrix.
- **Step 5:** Comparing the hypothetical power consumption values with the measured power traces. The Pearson's Correlation Coefficient equation is used as the statistical function to compare each key hypothesis's hypothetical power consumption values with the actual measured trace data at every position along the trace length. All statistical comparison results can be arranged in a

matrix of size $\boldsymbol{K}\times\boldsymbol{T}$, where indices of each element respectively indicate the corresponding hypothesis subkey value and position of the involved data point in the power trace. The indices of the element with the highest value in the comparison result matrix will show the hypothesis subkey that is most likely to be correct and the moment at which the chosen intermediate value has been processed. The Pearson's Correlation Coefficient can be calculated by using (3), as shown at the bottom of the page. The denotations in (3) are:

D is the total number of measured power consumption traces.

$H_{d,i}$ is the hypothetical power consumption value based on the d^{th} plaintext and i^{th} key hypothesis.

$T_{d,j}$ is the j^{th} sample point of the d^{th} measured trace.

$r_{i,j}$ is the correlation coefficient between the hypothetical power consumption value related to the i^{th} key hypothesis and the measured power traces at the j^{th} sample point.

C. TEST VECTOR LEAKAGE ASSESSMENT

Conducting practical CPA attacks is a reliable method to estimate the security of a targeted device. However, if the targeted device is protected with countermeasures against power analysis attacks, the number of traces required to reveal the secret key successfully would dramatically increase. That means the amount of data that needs to be analyzed is enormous, and this analysis process would take too many times to finish. In 2011, Goodwill *et al.* introduced a conformance style side-channel leakage testing methodology named Test Vector Leakage Assessment (TVLA) [51]. Later on, Cooper *et al.* reported that performing TVLA testing for side-channel leakage is faster by one to two orders of magnitude compared to conducting practical power analysis attacks [52]. The approach of TVLA was based on measuring the power consumption traces from the targeted device while executing cryptographic operations on a predefined set of input test vectors and then using statistical hypothesis testing to detect whether there is a sensitive intermediate value that significantly influences the measured traces or not. In [51] various categories of leakage tests are suggested, which focus on specific leakage points. However, Cooper *et al.* showed that the non-specific test (also known as the fixed versus random test) is the most powerful. This testing scheme can detect a wide variety of leakages with an order of magnitude fewer measurement traces than tests focusing on specific leakage points. Therefore, the non-specific test is used in many recent articles to evaluate the security of side-channel attacks resisted devices [8], [10], [13], [53].

In this work, we also use the non-specific TVLA test to evaluate the effectiveness of the proposed designing method. According to [52], details of the non-specific TVLA test procedure can be described as follows.

- **Step 1:** Acquiring power consumption traces.

Two sets of power traces (DataSet-1 and DataSet-2) must be collected by measuring the targeted device's power consumption while executing AES-128 encryption with a specific encryption key and a set of plaintext input.

For obtaining the DataSet-1, the encryption key is fixed to 0x0123456789ABCDEF123456789ABCDEF0. Then, $2n$ encryptions are performed on plaintext inputs $I_0, I_1, \ldots, I_{2n-1}$ with $I_{i+1} = AES - Encrypt(I_i)$; $i \in (0, 2n-1)$ and $I_0 =$ 0x00000000000000000000000000000000.

For obtaining the Dataset-2, the encryption key is also fixed to 0x0123456789ABCDEF123456789ABCDEF0. However, n encryptions are performed on fixed plaintext input $I_0 =$ 0xDA39A3EE5E6B4B0D3255BFEF95601890.

Acquiring power traces for two data sets are randomly interspersed to avoid any possible biased error due to measuring environments such as temperature or electromagnetic interference noise. The length of each power trace can be denoted as L.

- **Step 2:** Computing Welch's t-test [54] on first $n/2$ traces from DataSet-1 and first $n/2$ traces from DataSet-2. The Welch t-test equation is shown in (4).

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}} \tag{4}$$

  $\overline{X_1}$ and $\overline{X_2}$ are the sample means of two corresponding sub-datasets. S_1, S_2 are the sample standard deviations. N_1, N_2 are the number of traces in each subset of power traces, which are equal to $n/2$. As a result, a t-score trace T_{1st} of length L is obtained.

- **Step 3:** Computing Welch's t-test on second $n/2$ traces from DataSet-1 and last $n/2$ traces from DataSet-2. The same test in Step 2 is applied to different sub-datasets. As a result, another t-score trace T_{2nd} of length L is obtained. Repeating Welch's t-test twice on different data is necessary to minimize false positives in leakage detections [51].
- **Step 4:** Comparing two t-score traces T_{1st} and T_{2nd}. If both traces have a point that exceeds the ± 4.5 range at the same time during the middle third of the AES operation, the targeted device fails.

It is important to note that Welch's t-test is used to test the null hypothesis that the two sets of power traces (fixed-inputs set and random-inputs set) have identical means and variance.

$$r_{i,j} = \frac{D\sum_{d=1}^{D} H_{d,i}T_{d,j} - \sum_{d=1}^{D} H_{d,i}\sum_{d=1}^{D} T_{d,j}}{\sqrt{((\sum_{d=1}^{D} H_{d,i})^2 - D\sum_{d=1}^{D} H_{d,i}^2)((\sum_{d=1}^{D} T_{d,j})^2 - D\sum_{d=1}^{D} T_{d,j}^2)}} \tag{3}$$

In other words, the tester cannot detect any sensitive intermediate value that influences the measured power traces. A high absolute value of t-score indicates a high degree of confidence that the null hypothesis is incorrect. The ±4.5 limit is chosen so that if the t-score exceeds that limit, the null hypothesis can be rejected with 99.99% confidence. If the computed t-score maintained in the ±4.5 range, the targeted device is considered safe from power analysis attacks up to n power traces.

D. DEEP LEARNING BASED SIDE-CHANNEL ANALYSIS ATTACKS

In 2016, Maghrebi *et al.* introduced the Deep Learning-based Side-Channel Analysis (DL-SCA) attack as a state-of-the-art profiled power analysis attack [21]. They demonstrated that the DL-SCA is more powerful since it can easily break unprotected AES implementations and protected AES implementations with masked countermeasures. Since then, there have been various related works showing more advanced features of the DL-SCA attacks [22]–[24]. In [22], Cagli *et al.* successfully verified that DL-SCA attacks could be used against jitter-based countermeasures. In [23], Benadjila *et al.* introduced a profiled DL-SCA attack utilizing a convolutional neural network (CNN) that is suitable for attacking highly desynchronized power traces. In [24] demonstrated that DL-SCA attacks could also be applied in non-profiled scenarios.

In [23], Benadjila *et al.* conducted numerous experiments to experimentally validate the hyper-parameters selection for different neuron network models used in profiled DL-SCA attacks. Their work also compared the efficiency of four neuron network models when attacking desynchronized power consumption traces. These four models are *VGG*-16, *PCA-QDA*, MLP_{best}, and CNN_{best}. Their experimental results showed that the CNN_{best} model outperforms all other three models while requiring only a smaller number of training epochs. The CNN_{best} model also maintains the attack efficiency with high desynchronization in power traces. In this work, we propose to randomly scale the clock frequency of the hardware cryptographic accelerator in a wide range. This proposed countermeasure will introduce severe misalignment of POIs in measured power traces. A neuron network model capable of attacking highly desynchronized power traces is required to evaluate the proposed countermeasure's efficiency properly. Hence, we chose to employ the CNN_{best} architecture introduced in [23] in later experiments.

Performing a profiled DL-SCA attack consists of the following steps.

- **Step 1:** The adversaries must have full access to a profiling device, which is an open, identical copy of the targeted device. The profiling device's power traces are recorded while different inputs are encrypted/decrypted with different secret keys.
- **Step 2:** Each trace in the profiling traces is labelled using its corresponding input and secret key.
- **Step 3:** A Deep Learning training process is performed using the labelled traces as the training data. The result of this process is a trained convolutional neural network that can classify each power trace based on the corresponding key used.
- **Step 4:** The attackers measure another set of power traces from the targeted device. This set is called attacking traces. Then, the attackers use the trained network, obtained from step 3, to recover the secret key value used by the targeted device based on the network's classification results.

Similar to CPA attacks, each DL-SCA attack also targets only one byte of the secret key (*i.e.* a subkey) at a time. Therefore, the above steps must be repeated 16 times when trying to recover the full 16-byte secret key used by an AES-128 implementation.

E. CLOCK MANAGERS ON XILINX FPGAs - MMCM

The Mixed-mode Clock Manager (MMCM) is a Xilinx FPGA primitive, which can be used to synthesize a wide range of output clock frequencies from a fixed input clock signal. It can also work as a jitter filter for any clock signal. In the Kintex-7 FPGA series, which we used to implement our RISC-V SoC design, there are eight MMCMs. An MMCM primitive is a combination of both digital circuits and digital circuits. Hence, it can be dynamically reconfigured to change the output clock's frequency, phase shift, and duty cycle by writing suitable values to its controlling registers. Figure 2 illustrates the block diagram of an MMCM primitive. The MMCM primitive can generate up to seven clock outputs simultaneously. The input reference clock is passed through a programmable counter divider (D). Counter D is an integer counter that counts in the range of 1 to 106. The phase and frequency of the input reference clock are compared with that of the feedback clock by the Phase-Frequency Detector (PFD). The PFD would generate a control signal for the charge pump (CP) and loop filter (LF). This control signal is proportional to the difference between the phase and frequency of the input clock and the feedback clock. In turn, the CP and LF generate a reference voltage to the Voltage Control Oscillator (VCO). The VCO generates a high-frequency clock signal and passes it to eight programmable output counters ($O_0, O_1, \ldots, O_6$, and M). Counters $O_0, O_1, \ldots, O_6$, serve as frequency dividers and produce seven output clocks. The counter M produces a different clock signal that is used as a feedback clock. The counters $O_1, \ldots, O_6$, are integer counters that can have values in the range of 1 to 128, while the counters M and O_0 are fractional counters with 0.125 increments that can have values in the range of 2 to 64 and the range of 2 to 128, respectively. It means that a finer-scaled output clock frequency can be generated at the output of O_0, compared to other outputs of the MMCM primitive. The VCO also offers eight fixed phase variants and one variable phase variant of the generated high-frequency clock signal for fine-phase shifting. The MMCM is also integrated with other registers to control each final output clock signal's phase and

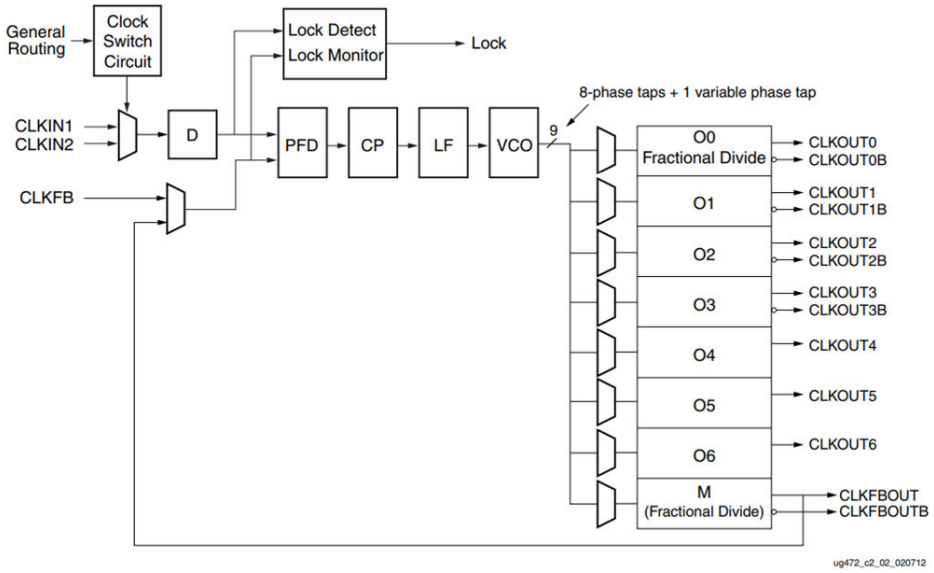

FIGURE 2. The block diagram of MMCM [55].

duty cycle. However, in this work, we are only interested in the frequency of the output clock signals and leave the phase and duty cycle reconfiguring features untouched.

Even though all programmable counters integrated into the MMCM primitive support a wide range of values, some restrictions must be followed. Xilinx recommends these restrictions to guarantee the stable operation of the MMCM, and they can be found in the Data Sheets of the corresponding FPGA chip. In this work, we implement the designed RISC-V SoC in a Kintex-7 FPGA. Therefore, the related limitation can be found in the Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics [56]. According to [56], the following restrictions must be followed:

- The MMCM primitive's input frequency F_{in} must be in the range of 10 MHz to 800 MHz.
- The input signal of the PFD, $F_{PFD} = \frac{F_{in}}{D}$ must be in the range of 10 MHz to 450 MHz.
- The output frequency of the VCO F_{VCO} must be in the range of 600 MHz to 1200 MHz.

When all of these restrictions are applied, the frequency of MMCM's output clock signals can be computed using (5).

$$F_{CLKOUT_i} = \frac{F_{in} \times M}{D \times O_i}; i \in (0, 6) \tag{5}$$

III. PROPOSED DESIGN

In order to improve the resistance of the targeted RISC-V SoC against power analysis attacks while still maintaining low overheads and minimizing the overall SoC's performance reduction, we propose applying the RDFS only to the targeted cryptographic accelerator, which is the AES128/256 MMIO peripheral in this case. We also utilize the Clock Managers primitive to generate the scaled clock frequencies dynamically. However, unlike previous works [8], [10], [13], where multiple output clock signals of the MMCM primitive are used simultaneously. These output clock signals are only generated with the integer counter mode. In our proposed method, we only employ the single fractional output clock signal to obtain a significant increase in the number of available scaled clock frequencies. Furthermore, we also do not alter the clock frequency after each clock cycle of the cryptographic operation. Instead, the stable scaled clock signal that drives the cryptographic accelerator is changed only after each encryption or decryption. By doing so, we can also guarantee that the misalignment of the POIs in each measured power trace corresponds to a distinct clock frequency. The number of possible POI's positions is precisely equal to the number of different clock frequencies generated. Lastly, we utilize the SoC's DDR memory to store all possible sets of parameters required to reconfigure the MMCM primitive dynamically. These parameters were previously synthesized and stored in the FPGA's Block RAM in previous related works. Thus, only a small number of reconfiguration parameters can be employed at a time due to the limitation of FPGA's

Some modifications need to be made to the targeted RISC-V SoC in order to deploy the proposed countermeasure. Figure 3 described the modified RISC-V SoC's architecture. First, a peripheral named Dynamic Reconfiguration Port (DRP), an additional MMCM primitive, and a pulse counter are added to the RISC-V SoC. Besides the 50 MHz clock signal F_{sys} that is used to drive the whole original system, the MMCM #1 also generates the F_{in} clock of 800 MHz. Furthermore, the TileLink connection between the AES accelerator and the Peripheral bus is changed to the asynchronous crossing type to allow the AES accelerator to operate in different clock domains with the rest of the system. The DRP is attached to the Peripheral bus as an MMIO peripheral, using the TileLink protocol. The DRP consists of several

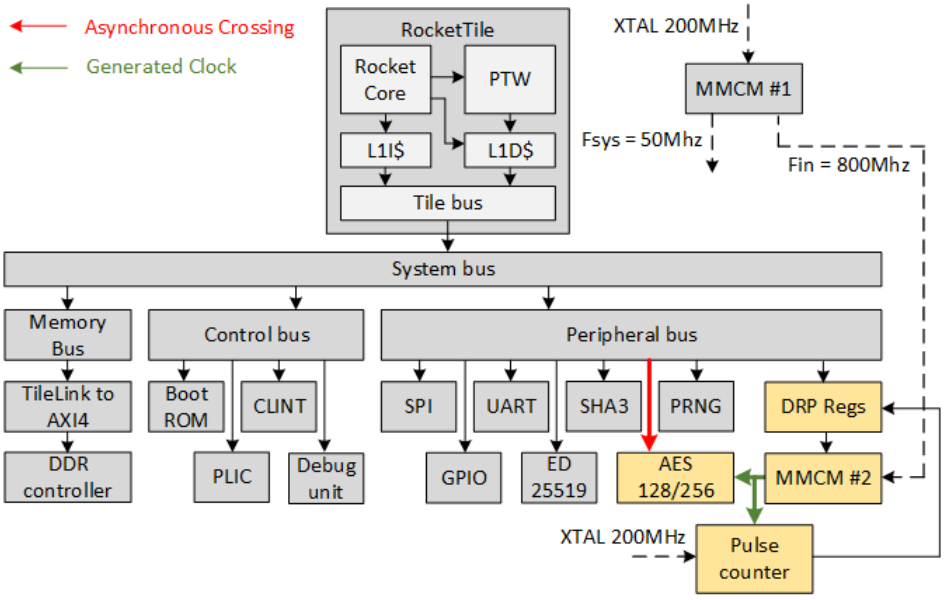

FIGURE 3. The modified RISC-V SoC architecture.

addressed registers and a finite state machine. The values stored in the addressed registers are the integer and fractional parts of the D, M, and O_0 counters. The finite state machine is open-source and provided in Xilinx's Application Note [57]. It derives the reconfiguration settings from the addressed registers' value and applies them to the MMCM #2. The MMCM #2 generates the new clock frequency based on the received configuration and the input clock F_{in}. The generated clock is used to drive the AES accelerator. The pulse counter also takes the generated clock as an input. It will count the number of the generated clock's rising edge in a fixed time interval, determined by the external 200 MHz reference clock. The counted result is forwarded to the addressed registers of the DRP peripheral. The RISC-V core can write the desired D, M, and O_0 values to dynamically change the AES accelerator's operating clock frequency without resetting the whole SoC. It can also read back the counted values to check the accuracy of the generated clock frequency.

At this point, the modified system described in Figure 3 can change the operating frequency of only the AES accelerator dynamically. To effectively use this ability to counter the power analysis attacks, we need to generate as many distinct clock frequencies as possible and apply the generated distinct clock frequencies to the AES accelerator after each encryption/decryption.

Since the MMCM primitive can be reconfigured during operation and the output clock frequency can be derived by using (5), we use a Matlab script to investigate the effects of F_{in} on the number of distinct output frequencies that can be generated. The Matlab script takes all the constraints of F_{in}, F_{PFD}, F_{VCO}, D, M, O_0 [55], [56] into account. It also considers the AES accelerator's maximum operating frequency reported in [50] as the upper limit for MMCM-generated output frequency. Besides, we want to minimize the time overhead of the proposed countermeasure. Therefore, the Matlab script also uses the overall system's operating frequency of 50 MHz as the lower limit for MMCM-generated output frequency. By testing the accuracy of the generated clock frequency with the added pulse counter, we also verify that the precision of the generated clock is ± 1 Hz. Figure 4 shows the relationship between MMCM's input clock frequency and the number of distinct output clock frequencies that can be generated. With an input frequency of 800 MHz, the MMCM can generate 219,412 distinct output clock frequencies. Therefore, we modified the MMCM #1 to generate an additional output clock signal F_{in} of 800 MHz. The Matlab script also produces a C header file containing all 219,412 combinations of D, M, and O_0 counters, corresponding to 219,412 distinct output clock frequencies. Each combination is represented by five bytes. One byte represents the integer D counter. Two bytes represent the fractional M counter, one for the integer part and the other for the fractional part. Similarly, the O_0 also requires two bytes to be represented. Therefore, all 219,412 different combinations can be represented by approximately 1.05MB.

Finally, a control program is composed to manage the countermeasure's operation. 1.05MB of pre-computed MMCM configurations are included in the program. The program and all MMCM configurations are compiled and stored in the execution memory space (the DDR RAM) during runtime. The program flowchart is illustrated in Figure 5. Before each AES encryption/decryption, the PRNG generates a random number. The random number is checked whether it is smaller than the number of MMCM configurations or not, which is 219,412. If it is larger than 219,412, the number is discarded, and the PRNG generates another random number.

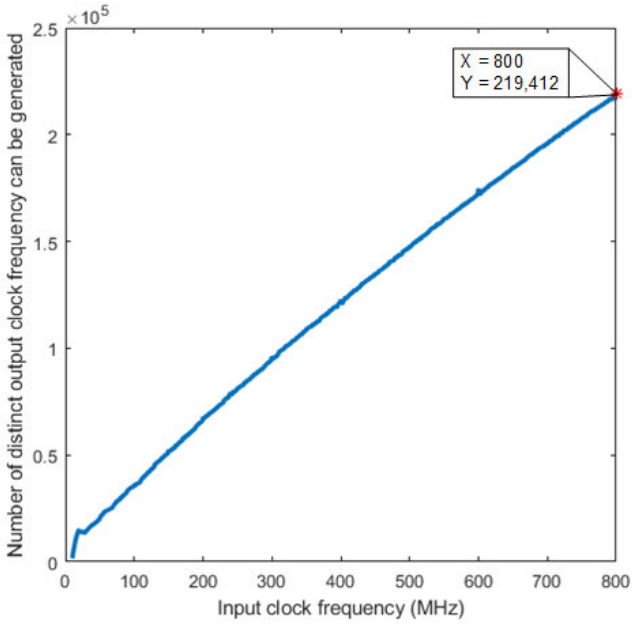

FIGURE 4. Number of generated output clock frequencies vs. Input clock frequency F_{in}.

This process is repeated until a number that is smaller than 219,412 is generated. The control program uses this number as an index and extracts the corresponding MMCM configuration. All related D, M, and O_0 counter's values are written to the DRP peripheral to generate and apply the new clock frequency to the AES accelerator. Once the MMCM asserts the Lock signal, indicating that the generated clock frequency is stable, the AES accelerator starts the encrypting/decrypting process as usual.

IV. SECURITY EVALUATION

A. EXPERIMENTAL SETUP

An experimental system is set up to acquire power traces automatically. Figure 6 illustrated our experimental system. It consists of a Monitoring PC, a Tektronix MSO2024B Oscilloscope, and the Sakura-X FPGA board as the test platform. The Sakura-X FPGA board features two separate FPGA chips, a Xilinx's Kintex-7 XC7K160T and a Xilinx's Spartan-6 XC6SLX45. A shunt resistor and probe points on the core VDD line of the Kintex-7 FPGA are also provided. Therefore, we implement the whole targeted RISC-V SoC into the Kintex-7 FPGA and measure the fluctuation of its logic core's VDD as the power traces. In this work, we grab a signal from the status register of the AES accelerator and map it into an FPGA's pin. This signal indicates the start of the AES encryption/decryption, and it will be used as the trigger signal for Oscilloscope's acquisitions.

The Tektronix MSO2024B oscilloscope is used to measure the power traces when the targeted RISC-V SoC processes AES-128 encryptions. This oscilloscope features four analog channels. The maximum bandwidth is 200 MHz, and the maximum sampling rate is 1 GS/s. Two passive probes are used for the measurement. One probe is used to detect

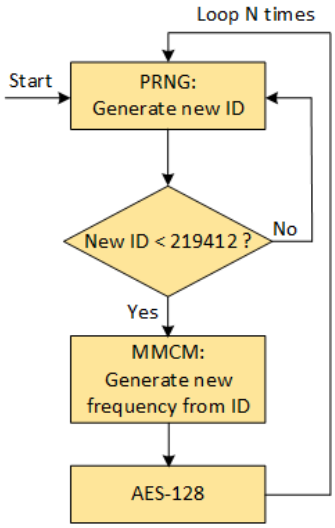

FIGURE 5. Flowchart of control program.

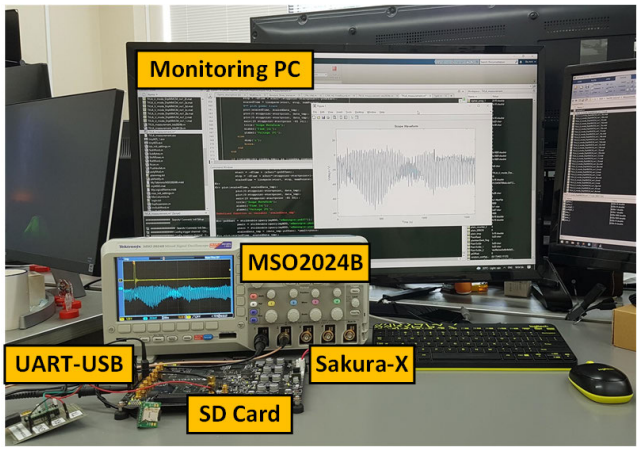

FIGURE 6. Experimental system.

the trigger signal provided by the target SoC. The other probe acquires the analog signal from the core V_{DD} node of the Kintex-7 FPGA. The Monitoring PC can remotely control this oscilloscope via VISA Virtual Instrument Software Architecture (VISA) connectivity.

The Monitoring PC is used to operate the whole auto-measuring system. It communicates with the oscilloscope through the USB port and with the targeted SoC through the UART port. It repeatedly sends plaintexts to the targeted SoC and commands the oscilloscope to acquire the power traces when the targeted SoC executes each encryption. After each encryption, the Monitoring PC receives the measured power trace from the oscilloscope and corresponding ciphertext from the targeted SoC. The Monitoring PC also verifies the ciphertext to ensure that the targeted SoC encrypts the plaintext correctly. The power trace and the corresponding plaintext and ciphertext are saved for later analysis.

This study employs a Monitoring PC equipped with an Intel i9-9820X CPU operated at 3.30 GHz and 96 GB of RAM. The Monitoring PC executes our self-developed

TABLE 1. Post-implementation utilization in Xilinx's Kintex-7 FPGA.

	Available	Original SoC		AES Accelerator		Protected SoC		Hardware
		Utilization	Utilization (%)	Utilization	Utilization (%)	Utilization	Utilization (%)	Overhead (%)
LUT	101400	48989	48.31	3169	3.13	51047	50.34	4.20
FF	202800	39298	19.38	3307	1.63	39516	19.49	0.55
BRAM	325	30	9.23	0	0.00	30	9.23	0
MMCM	8	2	25.00	0	0.00	3	37.50	50

MATLAB scripts to operate the auto-measuring system, perform the TVLA tests, and perform the CPA attacks. In the profiled DL-SCA attacks, the training and attacking processes are executed by a server equipped with two NVIDIA GeForce RTX 3090 GPUs. The source code of the training and attacking processes is written in Python, published by Benadjila *et al.* [23], and available at their GitHub repository [58].

B. EXPERIMENTAL RESULTS

1) IMPLEMENTATION RESULTS

First, the original RISC-V SoC design is generated and implemented into the Kintex-7 XC7K160T FPGA on the Sakura-X FPGA board. The maximum clock frequency that the implemented SoC can operate with is 50 MHz. An MMCM primitive generates the 50 MHz clock signal for the SoC system from an external 200 MHz crystal oscillator. After that, the modified RISC-V SoC design described in Section III is implemented into the Kintex-7 XC7K160T FPGA, replacing the original, unprotected one. The pulse counter, which checks the generated clock's accuracy in the designing process, is now unnecessary and is removed. Table 1 gives the post-implementation utilization results of both RISC-V SoCs. The integrated AES128/256 accelerator occupied only 3.13% of the total available Look-up table (LUT) and only 1.63% of the total available Flip-flop (FF) in a Kintex-7 XC7K160T FPGA chip. Meanwhile, the whole unprotected SoC requires 48.31% and 19.38% of the LUT and the FF, respectively. In other words, the AES accelerator only composes less than 8.5% of the overall SoC's hardware utilization. Furthermore, the table shows that with the additional hardware for applying the RDFS countermeasure, there is only a 4.2% and 0.55% increment in the number of used LUT and FF, respectively. The protected design also uses an extra MMCM primitive.

We also measured the execution times of 10 million AES-128 encryptions from both protected and unprotected designs. These encryptions are performed by the corresponded AES accelerators. The unprotected design's AES accelerator operates with a system clock of 50 MHz, while that of the protected design operates with a clock signal that is randomly chosen from 219,412 distinct clock frequencies, in the range of 50 MHz to 100 MHz. The measurement results show that the average time overhead is 3.36 times. Even though the AES accelerator was operated at higher frequencies, there are still timing penalties due to generating random indices and corresponding clock frequencies. In this work, we utilize a simple Linear Feedback Shift Register (LFSR) based PRNG. This PRNG is integrated as a peripheral to generate random values that are later used by other cryptographic peripherals (SHA3, Ed25519). The time overhead would be improved if another dedicated PRNG is integrated just for the RDFS countermeasure, like in other related works [8], [10], [13]. If the timing penalties due to generating the random indices are not considered, the time overhead is only 1.83 times.

2) TEST VECTOR LEAKAGE ASSESSMENT

First, the TVLA tests are performed on the unprotected RISC-V SoC. Power traces used for the TVLA tests are obtained as described in subsection II-C. Figure 7 shows the two TVLA test results with two disjoint groups of power traces. Each group contains 5,000 power traces for DataSet-1 and 5,000 power traces for DataSet-2. In both tests, nearly all of the t-score values exceed the ± 4.5 limit, and the absolute values peak at nearly 90, implying that the unprotected SoC is extremely vulnerable to power analysis attacks. When more power traces are added to the TVLA test, the absolute peaks become even higher. Figure 8 shows the test results with 50,000 power traces. The maximum t-score's absolute value increases to 180.

Subsequently, the TVLA tests are performed on the RISC-V SoC protected with our proposed RDFS countermeasure. However, as the first step, we acquire the power traces when applying only 1024 distinct frequencies to the AES accelerator. The TVLA test results in this scenario are described in Figure 9. A dataset of 10,000 power traces is used in this TVLA test. There are still some points in the t-score trace that have absolute values larger than 4.5, but the maximum t-score is significantly declined to around 14. We can observe that the RDFS countermeasure with only 1024 distinct frequencies has some limited effects in reducing the side-channel leakage.

Next, we apply the RDFS countermeasure using all 219,412 possible distinct clock frequencies. In both TVLA tests, each on five million power traces, there is no point in the t-score trace that exceeds the ± 4.5 limit during the middle third of the AES operation, as illustrated in Figure10. In other words, the SoC protected with 219,412 distinct clock frequencies passed the leakage test. Thus, we conclude that no information leakage can be detected in five million power traces.

3) CORRELATION POWER ANALYSIS ATTACKS

Realistic CPA attacks are conducted to verify the effectiveness of the proposed RDFS countermeasure. Power traces are obtained by using the experimental system described in

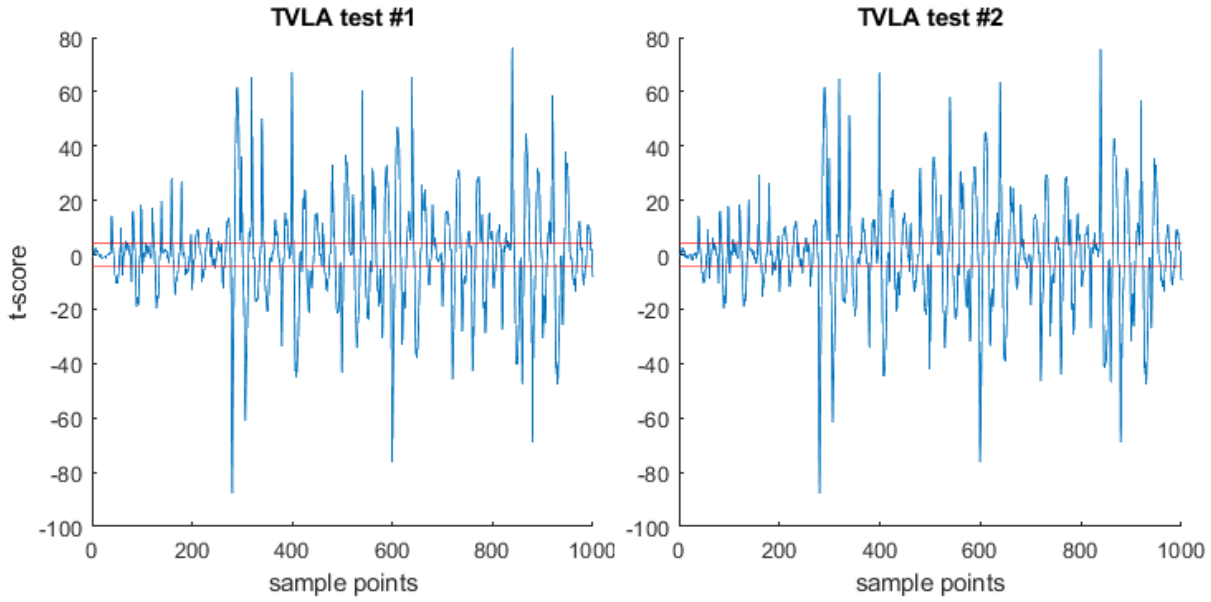

FIGURE 7. TVLA test on 10,000 traces measured from the unprotected SoC.

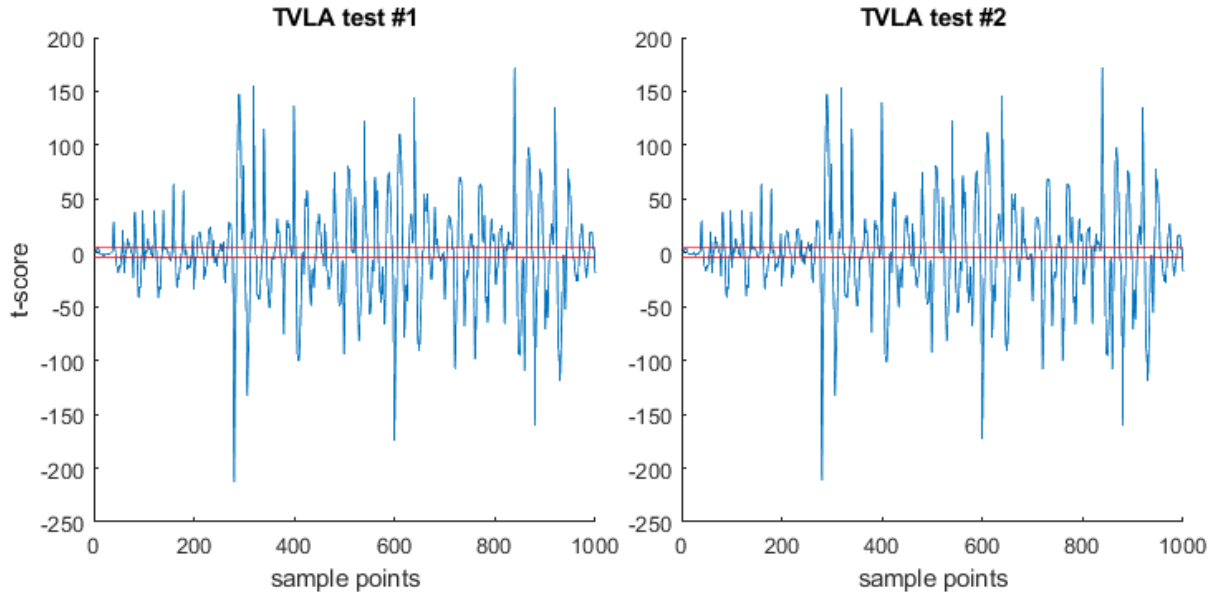

FIGURE 8. TVLA test on 50,000 traces measured from the unprotected SoC.

subsection IV-A. In this work, we perform multiple CPA attacks on the same set of measured power traces, targeting all 16 bytes of the secret encryption key used by the AES-128 accelerator. The Partial Guess Entropy (PGE) is used to evaluate the CPA attack's results [59]. It is the top-down ranking of the correct subkey when the statistical comparison results of all key hypotheses are sorted from highest to lowest. When the PGE reaches "0", the actual subkey has the highest statistical comparison result among all the key hypotheses, and the attack succeeds in recovering the subkey from the input set of power traces.

Figure 11 presents the results of CPA attacks on the unprotected SoC operating in bare-metal mode. Seventy thousand power traces are used in these attacks. The PGE results eventually reach "0" when attacking 12/16 subkeys, which means the CPA attacks failed to reveal four subkeys (byte numbers 2, 6, 10, and 14) while success with the other twelve subkeys. For each revealed byte, the minimum number of required traces varies from 1,642 traces to 58,685 traces, with an average of 28,636 traces. For the failed bytes, we also tried to perform CPA attacks using up to 100,000 traces. However, the PGE of these subkeys never reached "0".

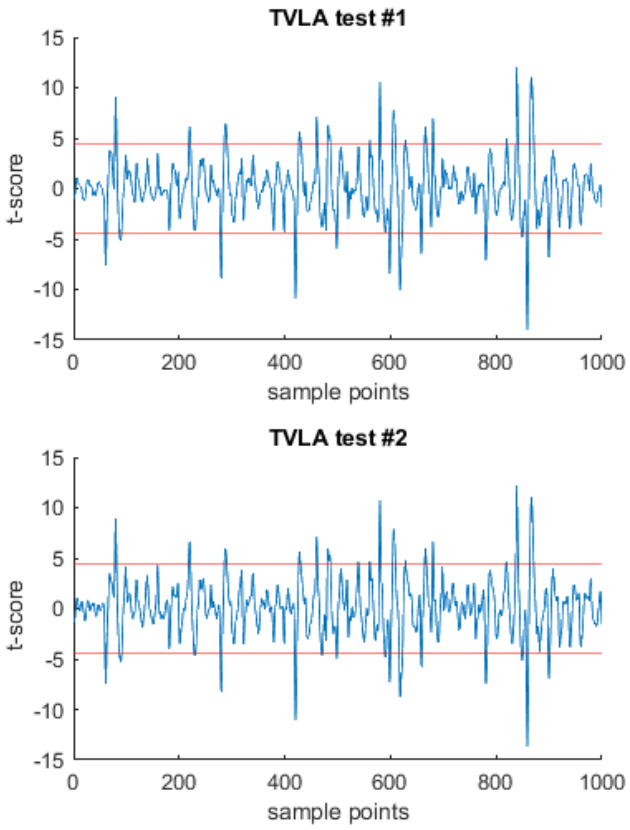

FIGURE 9. TVLA test on 10,000 traces measured from the protected SoC with 1024 distinct clock frequencies.

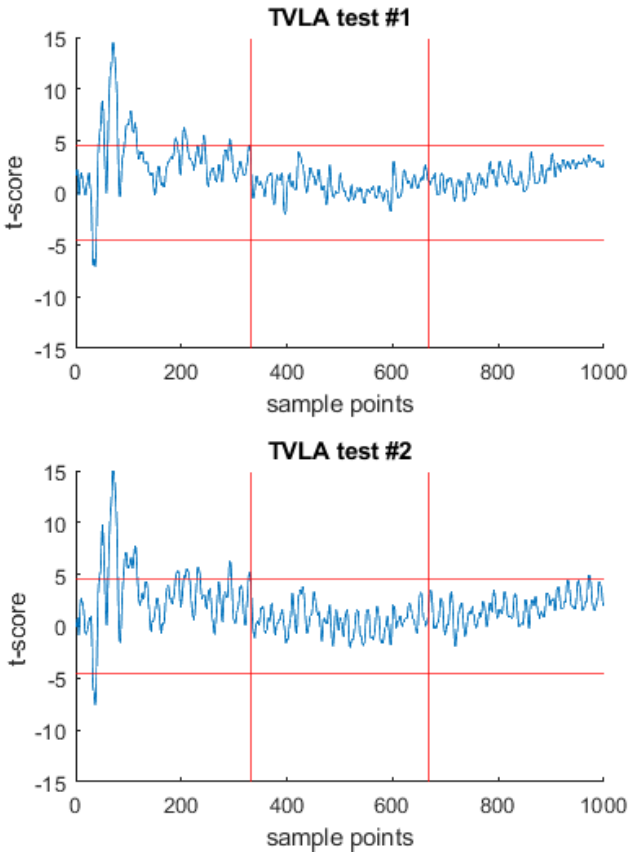

FIGURE 10. TVLA test on five million traces measured from the protected SoC with 219,412 distinct clock frequencies.

Since the AES accelerator's hardware utilization is less than 8.5% of the overall SoC's hardware utilization. These poor CPA attack results could be due to the measured power traces being too noisy. Therefore, we conduct other CPA attacks with a different set of power traces. This new set is measured using the averaging acquisition mode of the MSO2024B Oscilloscope. The experimental system described in subsection IV-A is modified. The monitoring PC sends a plaintext input to the targeted SoC 64 times consecutively. The Oscilloscope measures 64 power traces when the targeted SoC performs 64 encryptions on these same plaintext inputs. The average trace is calculated from these 64 traces and used in later CPA attacks. Average measurement could minimize the effects of random switching noise caused by the SoC's components other than the AES accelerator [60]. The results of CPA attacks on the unprotected SoC using averaged power traces are presented in Figure 12. Indeed, using averaged traces improves the CPA attacks' results. Thirteen subkeys are recovered successfully. The minimum number of required traces varies from 465 traces to 7,613 traces, with an average of 1,928 traces. However, these CPA attacks cannot obtain the correct values of three subkeys (byte numbers 6, 10, and 14). In practice, attackers might use brute force attacks on the last three subkeys and finally compromise the whole system's security.

Additionally, CPA attacks on the unprotected SoC operating in OS mode are also performed. Twenty thousand averaged traces are used in the analysis process. The corresponding results are provided in Figure 13. These attacks also reveal thirteen subkeys successfully. The remaining uncovered subkeys are byte numbers 6, 10, and 14, similar to attacking unprotected, bare-metal mode SoC. However, the minimum number of required traces for revealing other subkeys is increased since the power traces contain more noise in this case. The minimum number of required traces varies from 1,650 traces to 19,591 traces, with an average of 10,175 traces.

The CPA attacks results on unprotected SoC demonstrated that a cryptographic accelerator integrated into a complex SoC is vulnerable to power analysis attacks, even when its size is much smaller than the overall SoC's size. Besides, it was also confirmed that the TVLA could detect the side-channel leakage earlier than realistic attacks.

Next, we conduct the CPA attacks on the SoC protected with the proposed RDFS countermeasure, in which the AES accelerator's operating frequency is randomly altered between 219,412 distinct frequencies in the range of 50 MHz to 100 MHz. The targeted SoC still operates in bare-metal mode. The attack results are illustrated in Figure 14. It shows that even after analyzing five million traces of the

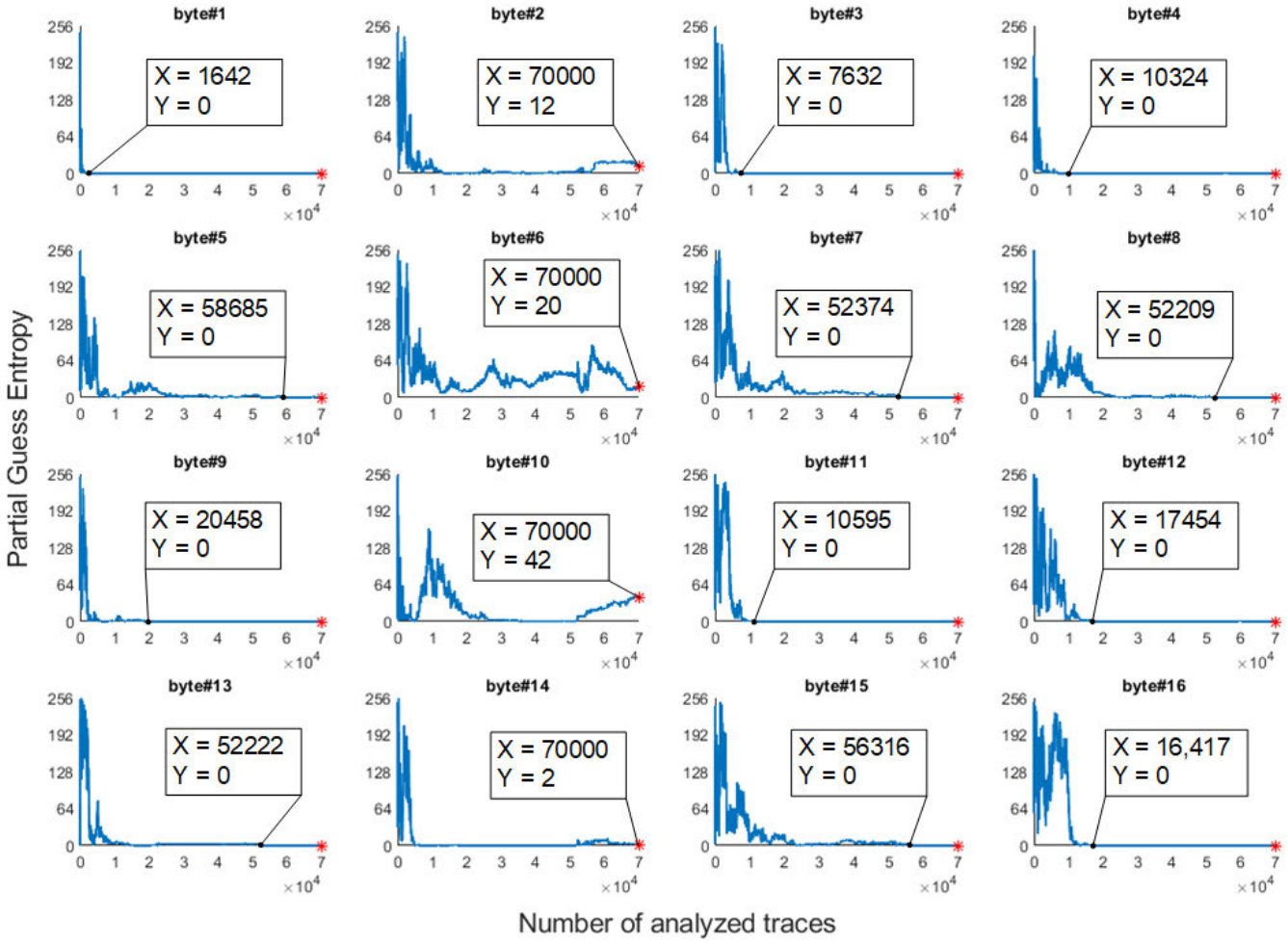

FIGURE 11. CPA attacks' results on 70,000 traces measured from the unprotected SoC operating in bare-metal mode.

protected SoC, none of the PGE results targeting 16 subkeys can reach "0". In other words, the CPA attacks failed to recover any secret subkey. Besides, the proposed RDFS countermeasure also prevents using the average measuring technique to conduct CPA attacks on the protected SoC because the clock frequency of the AES accelerator is changed after each encryption. Therefore, we can conclude that applying the RDFS countermeasure with 219,412 distinct frequencies will significantly enhance the CPA resistance of the targeted SoC. Moreover, since the TVLA test cannot detect any leakage with the dataset of five million traces, we believe that the proposed design can resist the CPA attacks with even more than five million power traces. Compared to attacking unprotected SoC, bare-metal mode and using average measuring, the number of traces required to attack protected SoC successfully skyrocketed from 1,928 traces to more than five million traces, which is a nearly 2,593-fold increase.

4) PROFILED DEEP LEARNING BASED SIDE CHANNEL ATTACKS

Recently, the state-of-the-art DL-SCA attacks were reported to outperform classic CPA attacks when attacking misaligned power traces. Therefore, we also evaluate the effectiveness of our proposed design against the DL-SCA attacks. Various profiled DL-SCA attacks utilizing the CNN_{best} network architecture [23] are performed on the previously measured power traces.

The attack parameters and results are summarized in Table 2. In each DL-SCA attack, the CNN_{best} model is trained with a fixed batch size and an epoch of 200 and 75, respectively. The results confirmed that the profile DL-SCA attacks are more powerful than the classic CPA attacks. The DL-SCA can completely recover the 16-bytes secret encryption key in attacks on the unprotected SoC operating in bare-metal mode. The average number of traces required from the targeted device is less than corresponding CPA attacks. However, DL-SCA attacks have worse performance than CPA attacks when targeting the unprotected SoC operating on the Linux OS. Only nine subkeys can be revealed. This result agrees with Alipour *et al.* [61] that a noise-generation-based hiding countermeasure may provide better protection against DLSCAs than other types of countermeasure. Unfortunately, the results show that the protected SoC utilizing the RDFS countermeasure with 219,412 distinct clock frequencies can

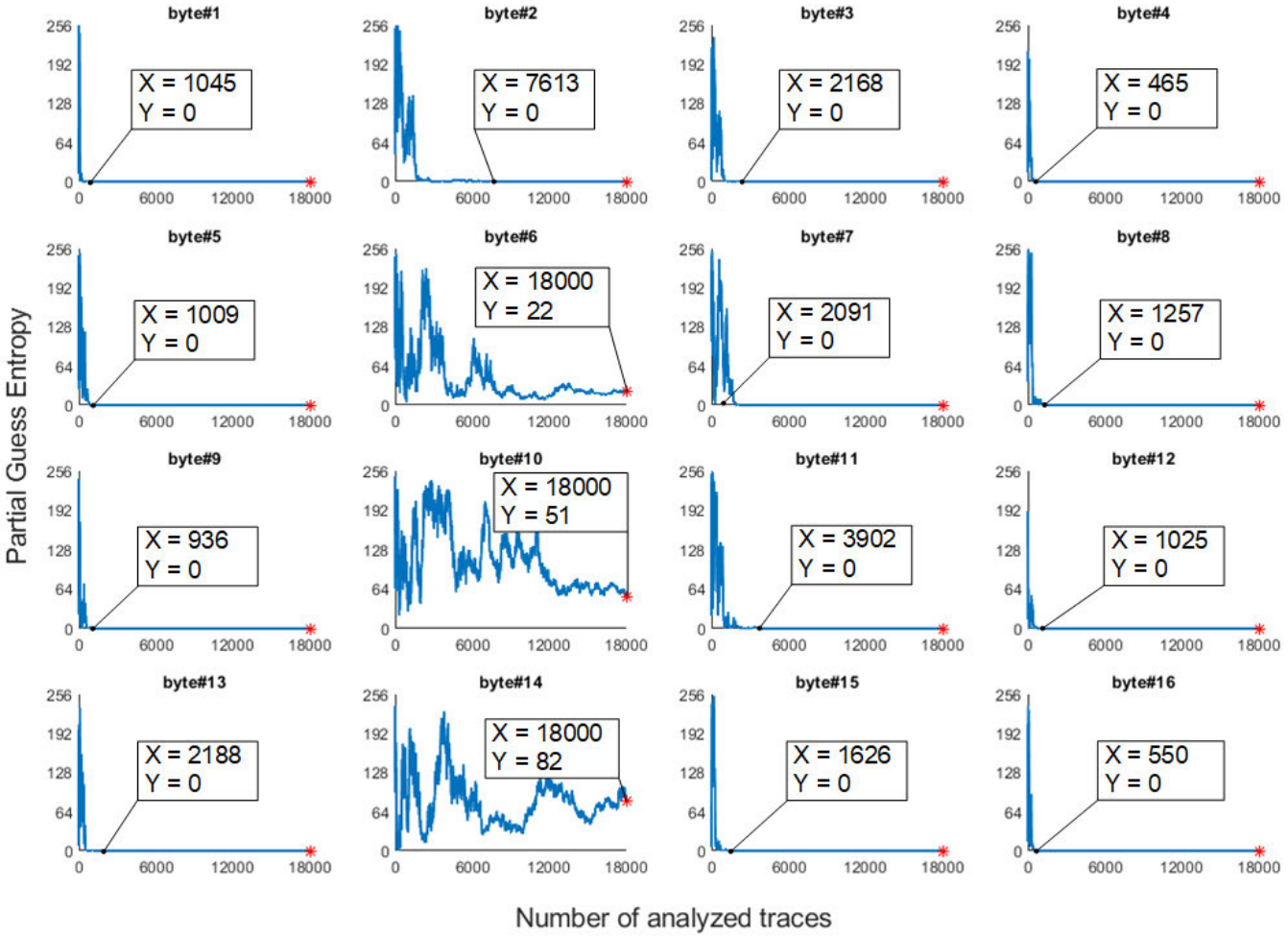

FIGURE 12. CPA attacks' results on 18,000 traces measured from the unprotected SoC operating in bare-metal mode. The power traces are measured in averaging mode.

TABLE 2. Profiled DL-SCA attacks' parameters and results.

DLSCA attack no.		#1	#2	#3	#4	#5
Parameters	Targeted device	Unprotected SoC	Unprotected SoC	Unprotected SoC	Protected SoC	Protected SoC
	Operating mode	Bare-metal	Bare-metal	Linux OS	Bare-metal	Bare-metal
	Traces measuring method	Single acquisiton	Averaging-64	Averaging-64	Single acquisition	Single acquisition
	Number of profiling traces	60,000	15,000	17,000	1,000,000	60,000
	Number of attacking traces	12,000	3,000	3,000	100,000	12,000
Results	Number of subkeys revealed successfully	16/16	16/16	9/16	13/16	0/16
	Minimum required traces (on average)	4,231	805	2,022	45,924	N/A

be defeated by the DL-SCA attacks. One million traces measured from the protected SoC are used as profiling traces, while another 100,000 traces are used as attacking traces. Thirteen subkeys are recovered successfully, with the minimum number of required traces is 45,924 traces on average. This result further demonstrates the power of DL-SCA against misalignment-based hiding countermeasures.

In the last DL-SCA attack experiment, the same number of traces were used for the profiling and attacking phases. In such a scenario, DL-SCA attacks failed to recover any subkeys. Therefore, we can conclude that even though the proposed RDFS countermeasure cannot completely prevent the DL-SCA attacks, it still helps to improve the protected device's resistance.

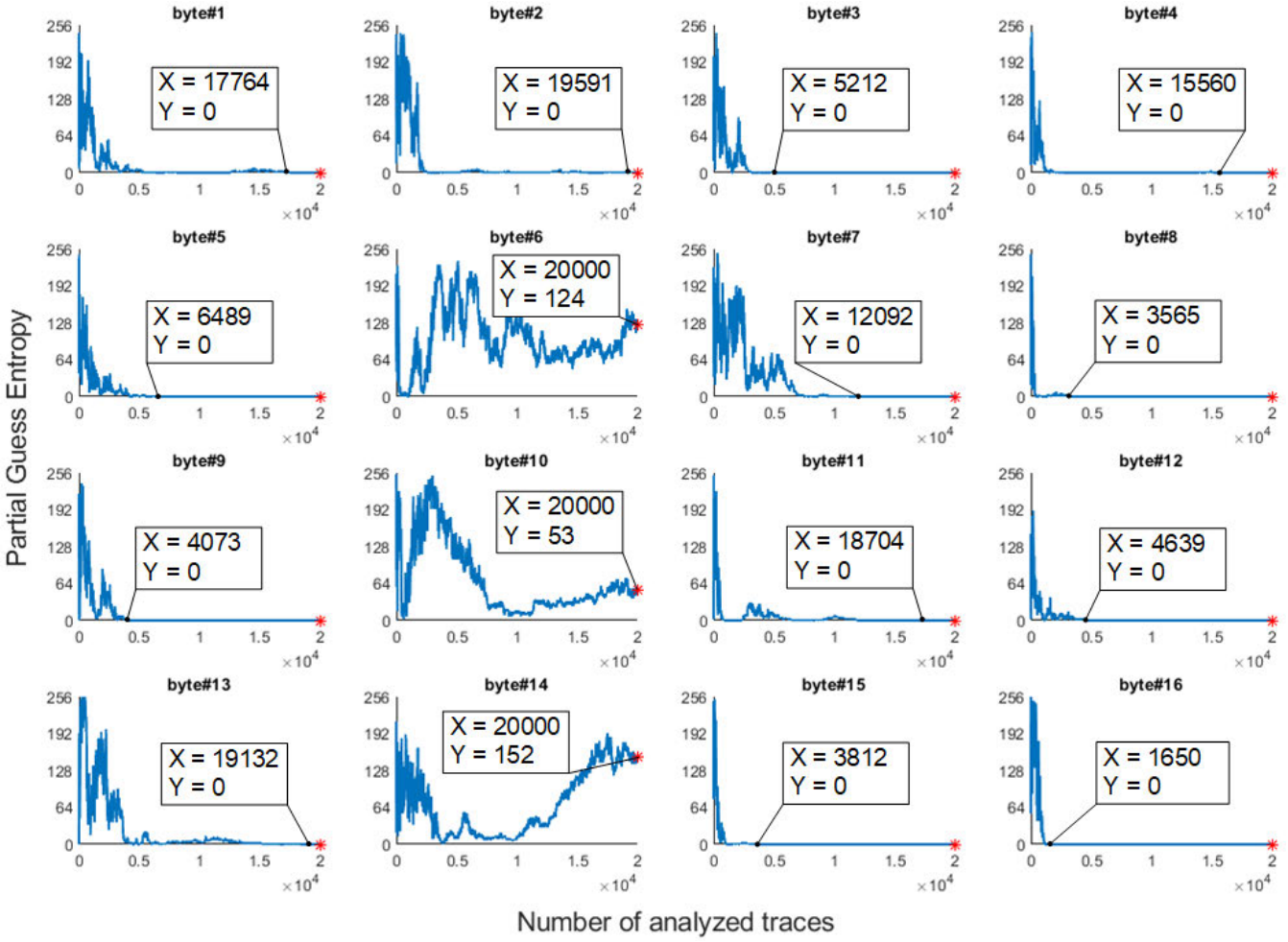

FIGURE 13. CPA attacks' results on 20,000 traces measured from the unprotected SoC operating in OS mode. The power traces are measured in averaging mode.

TABLE 3. Comparison with related works.

	Target	Counter-measure	Overhead			TVLA	Attacked bytes	CPA	DL-SCA	
			Timing	Power	Area				Parameters	Results
[53]	Multicore SoC (ASIC)	RTS, RIO, FPR, PSMC	1.04×	1.35×	1.23×	Failed @ 2×10^5	Only byte #0	2×10^6	Profiling: 1×10^6 Attacking: 1×10^5	Cannot reveal byte #0
[8]	RISC-V SoC (FPGA)	SCRIP	1.88×	N/A	LUT: 1.04× FF: 1.03× MMCM: 2×	Passed @ 2×10^5	N/A	3×10^5	N/A	
[13]	AES core (ASIC)	DFR-8	58.9×	1.01×	N/A	Passed @ 5×10^6	N/A	1×10^6	Profiling: 5×10^5 Attacking: N/A	Cannot reveal any byte
[10]	AES core (ASIC)	RFTC	1.72×	1.48×	1.3×	Passed @ 1×10^6	N/A	4×10^6	N/A	
This work	RISC-V SoC with AES accele-rator (FPGA)	RDFS	3.36 ×	1.06×	LUT: 1.042× FF: 1.0055× MMCM: 2×	Passed @ 5×10^6	All 16 bytes	5×10^6	Profiling: 1×10^6 Attacking: 1×10^5	13/16 subkeys

C. RESULTS COMPARISON

The effectiveness of our proposed technique is compared with other state-of-the-art randomizing-clock-frequency-based countermeasures. Table 3 summarized all the experimental results obtained from the previous subsection and provided a comparison of the results. Our proposed countermeasure is applied to a RISC-V SoC with an AES accelerator. The whole system is implemented on a Kintex-7 FPGA. Applying the proposed countermeasure only required implementing a few registers, an MMCM, and a lightweight, open-source finite state machine [57]. Therefore, this work achieved a lower area overhead compared to [8] with only 1.042 times

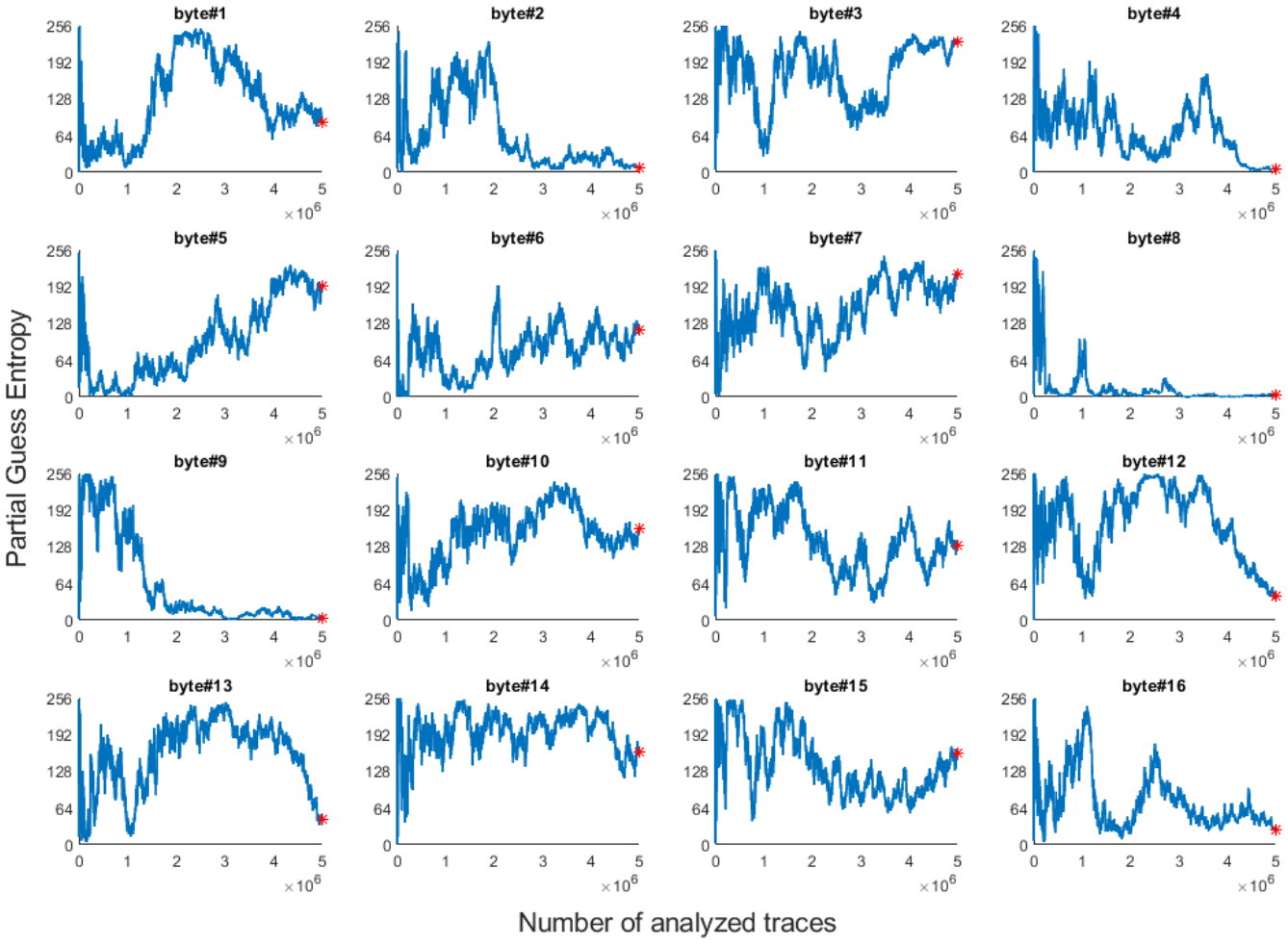

FIGURE 14. CPA attacks' results on five million traces measured from the protected SoC, targeting all 16 bytes.

and 1.0055 times increase in LUT and FF, respectively. Both [8] and this work employ an additional MMCP to generate the new clock frequency, despite the fact that both of the corresponding original target devices already have one MMCM. Hence, the numbers of MMCM are considered as doubled. Besides, even though ASIC and FPGA are two different technologies, the area overhead of this work is relatively lower than the area overheads reported in [53] and [10], which are 1.23 times and 1.3 times, respectively. Moreover, the protected SoC only has the power overhead of 1.06 times. The power estimations of the unprotected SoC and the protected SoC are acquired from the synthesis report of Xilinx's Vivado tool. The power overhead is computed based on these estimations. Table 3 showed that the power overhead of the proposed countermeasure is better than that of [10], [53], but not as good as that of [13]. Meanwhile, the timing overhead of the proposed countermeasure is 3.36 times. It is calculated after measuring the execution times of 10 million AES-128 encryptions from both protected and unprotected designs. Table 3 showed that the timing overhead of the proposed countermeasure is significantly lower than that of [13], but it is still higher than the timing overhead of [8], [10], [53]. The timing overhead of the proposed design is contributed by the time intervals for selecting a random MMCM configuration, applying the selected configuration to the MMCM, and generating the corresponding clock frequency after each encryption/decryption. In [8], [10], [53], multiple clock signals are generated in advance and randomly switched to drive the targeted cryptographic core, hence their timing overhead is lower. In contrast, both [13] and this work chose to generate new clock signals with different frequencies during run-time, thus more timing overhead is required. Besides, this work used hardware modules (including the on-chip PRNG and DRP peripherals) to generate the new clock signals, while [13] used a software program to generate the necessary random parameters.

The protected SoC's resistance against power analysis attacks is evaluated by testing with the TVLA test, then conducting practical CPA attacks and DL-SCA attacks targeting all 16 bytes of the AES encryption key. The TVLA test with five million traces cannot detect any side-channel leakage from the protected SoC. When compared to other

related works, this is the best TVLA result. The RISC-V SoC equipped with our proposed countermeasure also withstands the classic CPA attacks at five million traces. Table 3 shows that our protected design achieves the best resistance when evaluated with CPA attacks and TVLA tests. The reasons for this achievement are twofold. First, we proposed to consider the misalignment of POIs as the most important designing parameter. Second, the proposed design is capable of generating an enormous number of 219,412 distinct clock frequencies. Therefore, changing the clock frequency after each encryption/decryption ensures 219,412 cases of POIs misalignment.

Unfortunately, the protected SoC is still defeated by the profiled DL-SCA attacks. In our experiments, one million power traces are used in the profiling phase and then the attacking phase is conducted on 100,000 traces. The DL-SCA attacks can reveal up to 13/16 subkeys. However, as described in subsection IV-B4, our proposed countermeasure still helps to improve the DL-SCA resistance when the same profiling and attack conditions are applied. In the case of DL-SCA, [13], [53] provided better solutions that DL-SCA could not defeat. In [53], Yang *et al.* proposed to combine several different countermeasures, including Random Task Scheduling (RTS), Random Insertion of Operation (RIO), Frequency and Phase Randomization (FPR), and Power State Monitoring and Control (PSMC). However, they only perform the security evaluation by attacking the first byte of the secret key. The DL-SCA attacks failed to reveal that first byte. The DL-SCA attack results for the remaining fifteen bytes are not provided. Moreover, if each countermeasure is applied separately, the DL-SCA still successfully reveals the secret subkey. In [13], Hettwer *et al.* also use the MMCM to dynamically alter the clock frequency of an AES core. DL-SCA attacks cannot defeat their countermeasure, but the cost of timing overhead is enormous. The encryption time of their proposed design increases nearly 59 fold. Besides, the targeted bytes of the secret key which are being attacked are not provided in [13].

V. FURTHER DISCUSSION

The experimental results described in the previous sections show the practical evaluation of the effectiveness of our proposed countermeasures against CPA and DL-SCA attacks. Using the MMCM primitive in Xilinx's FPGA, the countermeasure generates many different clock frequencies for driving the AES accelerator. The AES accelerator performs each encryption/decryption with a randomly chosen clock frequency. As a result, the resistance against power analysis attacks is significantly improved while low overheads are maintained. However, some limitations exist in our work. First, the targeted device used in our experiments is a cryptographic RISC-V SoC inherited from our previous work. There is only an LFSR-based PRNG attached to the SoC as an MMIO peripheral. Therefore, the random values for selecting the AES accelerator's operating frequency are only pseudorandom. An integrated true random number generator would improve the generated values' randomness, further enhancing resistance against power analysis attacks. Moreover, the PRNG is attached as an independent peripheral. Therefore, it also contributes additional time overhead since generating a random number must be done before generating the corresponding clock frequency and encryption. Having another dedicated random number generator would solve the problem and further reduce the time overhead. Our future works will be directed toward integrating an on-chip true random number generator, dedicated only to the countermeasure against power analysis attacks.

Moreover, applications of chaos theory in cryptography have recently attracted attention. These applications mostly include hash functions [62], [63], random number generation [64], [65], and image encryption algorithms [66]–[68]. Similar to conventional algebraic cryptography, chaos-based encryption/decryption algorithms are also prone to power analysis attacks. Their vulnerabilities are experimentally demonstrated in various recent works [69], [70]. El-Moursy *et al.* proposed using a chaos-based technique as a countermeasure against power analysis attacks for an AES processor in 2020 [71]. El-Moursy's countermeasure has a similar approach to the approaches of this work and other related papers mentioned in subsection I-A [8], [10], [13], [44]. In [71], chaotic clock signals derived from chaotic systems (*i. e.* the single-switch jerk chaotic oscillator (SSJSO) and the two-wing chaotic oscillator (TWCO)) are used to drive an AES processor and protect it from power analysis attacks. The authors showed that these chaotic systems could be implemented on chip at a very low hardware cost, which is far cheaper than applying a truly random RDFS solution. Even though the chaotic clock signals are not random, they are similar to random signals in being unpredictable. Hence, the positions of the POIs in the power traces acquired from the targeted AES processor are unpredictable and misaligned. Unfortunately, the experimental results provided by El-Moursy *et al.* are very limited. They demonstrated that the AES processor protected by their proposal could not be broken by analyzing 1,000 power traces. This figure is incomparable with those presented in this work and other related papers, ranging from a few hundred thousand up to several millions of power traces. In our opinion, El-Moursy's countermeasure is capable of reaching a higher level of protection against power analysis attacks. However, more extensive experimental evaluation is required before any comparison can be made.

VI. CONCLUSION

This article has demonstrated the vulnerability to different types of power analysis attacks of complex cryptographic SoC, even when the cryptographic components require only a minor proportion of the overall system's hardware utilization. Furthermore, a countermeasure against power analysis attacks for such complicated cryptographic SoC is proposed. Its effectiveness was evaluated by conducting practical CPA attacks, DL-SCA attacks and TVLA leakage tests.

The experimental results show that the RISC-V SoC protected with the proposed countermeasure cannot be broke by classic CPA attacks with more than five million power traces, which is an improvement of at least 2,593 fold. The protected RISC-V SoC also passed the TVLA leakage test at five million power traces, which is the highest figure comparing to other recent related works. The proposed countermeasure also helped to harden the targeted RISC-V SoC against the state-of-the-art DL-SCA attacks.

REFERENCES

[1] S. Feng, J. Wu, S. Zhou, and R. Li, "The implementation of LeNet-5 with NVDLA on RISC-V SoC," in *Proc. IEEE 10th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Oct. 2019, pp. 39–42.

[2] X. Zhong, C.-W. Sham, and L. Ma, "A RISC-V SoC for mobile payment based on visible light communication," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Dec. 2020, pp. 102–105.

[3] A. Arnaud, M. Miguez, J. Gak, R. Puyol, R. Garcia-Ramirez, E. Solera-Bolanos, R. Castro-Gonzalez, R. Molina-Robles, A. Chacon-Rodriguez, and R. Rimolo-Donadio, "A RISC-V based medical implantable SoC for high voltage and current tissue stimulus," in *Proc. IEEE 11st Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb. 2020, pp. 1–4.

[4] U. Banerjee, A. Wright, C. Juvekar, M. Waller, Arvind, and A. P. Chandrakasan, "An energy-efficient reconfigurable DTLS cryptographic engine for securing Internet-of-Things applications," *IEEE J. Solid-State Circuits*, vol. 54, no. 8, pp. 2339–2352, Aug. 2019.

[5] Z. Zang, Y. Liu, and R. C. C. Cheung, "Reconfigurable RISC-V secure processor and SoC integration," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 827–832.

[6] C. Duran, H. Gomez, and E. Roa, "AES sbox acceleration schemes for low-cost SoCs," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[7] 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.

[8] 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.

[9] R. Xu, L. Zhu, A. Wang, X. Du, K.-K.-R. Choo, G. Zhang, and K. Gai, "Side-channel attack on a protected RFID card," *IEEE Access*, vol. 6, pp. 58395–58404, 2018.

[10] 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.

[11] A. Singh, M. Kar, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "A 128b AES engine with higher resistance to power and electromagnetic side-channel attacks enabled by a security-aware integrated all-digital low-dropout regulator," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 404–406.

[12] X. Cai, R. Li, S. Kuang, and J. Tan, "An energy trace compression method for differential power analysis attack," *IEEE Access*, vol. 8, pp. 89084–89092, 2020.

[13] B. Hettwer, K. Das, S. Leger, S. Gehrer, and T. Guneysu, "Lightweight side-channel protection using dynamic clock randomization," in *Proc. 30th Int. Conf. Field-Program. Log. Appl. (FPL)*, Aug. 2020, pp. 200–207.

[14] Xilinx, San Jose, CA, USA. (2020). *UG1075 (V1.9) Zynq UltraScale+ Device Packaging and Pinouts, Product Specification User Guide*. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug1075-zynq-ultrascale-pkg-pinout.pd

[15] 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.

[16] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems—CHES 2002*, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds. Berlin, Germany: Springer, 2003, pp. 13–28.

[17] J. Doget, E. Prouff, M. Rivain, and F.-X. Standaert, "Univariate side channel attacks and leakage modeling," *J. Cryptograph. Eng.*, vol. 1, no. 2, p. 123, 2011.

[18] A. Chakraborty, B. Mazumdar, and D. Mukhopadhyay, "A practical DPA on grain v1 using LS-SVM," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2015, pp. 44–47.

[19] L. Duan, Z. Hongxin, L. Qiang, Z. Xinjie, and H. Pengfei, "Electromagnetic side-channel attack based on PSO directed acyclic graph SVM," *J. China Univ. Posts Telecommun.*, vol. 22, no. 5, pp. 10–15, Oct. 2015.

[20] S. Hou, Y. Zhou, H. Liu, and N. Zhu, "Wavelet support vector machine algorithm in power analysis attacks," *Radioengineering*, vol. 26, no. 3, pp. 890–902, 2017.

[21] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.* Hyderabad, India: Springer, 2016, pp. 3–26.

[22] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst.* Taipei, Taiwan: Springer, 2017, pp. 45–68.

[23] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas. (2018). Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. ANSSI, CEA, LETI, MINATEC Campus, Paris, France, [Online]. Available: https://eprint.iacr.org/2018/053.pdf

[24] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 2, pp. 107–131, Feb. 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/7387

[25] L. Benini, A. Macii, E. Macii, E. Omerbegovic, F. Pro, and M. Poncino, "Energy-aware design techniques for differential power analysis protection," in *Proc. 40th Conf. Design Autom.*, 2003, pp. 36–41.

[26] S. Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Y. Xie, "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach," in *Proc. Design, Autom. Test Eur.*, vol. 3, 2005, pp. 64–69.

[27] M. Kar, A. Singh, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "8.1 improved power-side-channel-attack resistance of an AES-128 core via a security-aware integrated buck voltage regulator," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 142–143.

[28] E. Laohavaleeson and C. Patel, "Current flattening circuit for DPA countermeasure," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Jun. 2010, pp. 118–123.

[29] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. 28th Eur. Solid-State Circuits Conf.*, 2002, pp. 403–406.

[30] K. Tiri and I. Verbauwhede, "Secure logic synthesis," in *Field Programmable Logic and Application*, J. Becker, M. Platzner, and S. Vernalde, Eds. Berlin, Germany: Springer, 2004, pp. 1052–1056.

[31] P. E. Andrews and M. S. Dhanesh, "A body biased adiabatic dynamic differential logic(BADDL) to prevent DPA attacks in smart cards," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2017, pp. 686–690.

[32] M.-L. Akkar and L. Goubin, "A generic protection against high-order differential power analysis," in *Fast Software Encryption*, T. Johansson, Ed. Berlin, Germany: Springer, 2003, pp. 192–205.

[33] T. S. Messerges, "Securing the AES finalists against power analysis attacks," in *Fast Software Encryption*, G. Goos, J. Hartmanis, J. van Leeuwen, and B. Schneier, Eds. Berlin, Germany: Springer, 2001, pp. 150–164.

[34] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology—CRYPTO'99*, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 398–412.

[35] L. Goubin and J. Patarin, "DES and differential power analysis the 'duplication' method," in *Cryptographic Hardware and Embedded Systems*, Ç. K. Koç and C. Paar, Eds. Berlin, Germany: Springer, 1999, pp. 158–172.

[36] Y. Desmedt, "Some recent research aspects of threshold cryptography," in *Information Security*, E. Okamoto, G. Davida, and M. Mambo, Eds. Berlin, Germany: Springer, 1998, pp. 158–173.

[37] E. De Mulder, S. Gummalla, and M. Hutter, "Protecting RISC-V against side-channel attacks," in *Proc. 56th ACM/IEEE Annu. Design Autom. Conf. (DAC)*, Jun. 2019, pp. 1–4.

[38] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni, "Fresh re-keying: Security against side-channel and fault attacks for low-cost devices," in *Proc. Int. Conf. Cryptol. Afr.* Stellenbosch, South Africa: Springer, 2010, pp. 279–296.

[39] S. Vuppala, A. E.-D. Mady, and A. Kuenzi, "Moving target defense mechanism for side-channel attacks," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1810–1819, Jun. 2020.

[40] Y. Gui, S. M. Tamore, A. S. Siddiqui, and F. Saqib, "Key update countermeasure for correlation-based side-channel attacks," *J. Hardw. Syst. Secur.*, vol. 4, no. 3, pp. 167–179, Sep. 2020.

[41] J. Zhang, C. Shen, Z. Guo, Q. Wu, and W. Chang, "CT PUF: Configurable tristate PUF against machine learning attacks for IoT security," *IEEE Internet Things J.*, early access, Jun. 18, 2021, doi: 10.1109/JIOT.2021.3090475.

[42] K. Baddam and M. Zwolinski, "Evaluation of dynamic voltage and frequency scaling as a differential power analysis countermeasure," in *Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID)*, 2007, pp. 854–862.

[43] B.-A. Dao, T.-T. Hoang, A.-T. Le, A. Tsukamoto, K. Suzaki, and C.-K. Pham, "Exploiting the back-gate biasing technique as a countermeasure against power analysis attacks," *IEEE Access*, vol. 9, pp. 24768–24786, 2021.

[44] T. Güneysu and A. Moradi, "Generic side-channel countermeasures for reconfigurable devices," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Nara, Japan: Springer, 2011, pp. 33–48.

[45] T.-T. Hoang, C. Duran, D.-T. Nguyen-Hoang, D.-H. Le, A. Tsukamoto, K. Suzaki, and C.-K. Pham, "Quick boot of trusted execution environment with hardware accelerators," *IEEE Access*, vol. 8, pp. 74015–74023, 2020.

[46] University of California at Berkeley. (2020). *Chipyard: An Agile RISC-V SoC Design Framework With in-Order Cores, Out-of-Order Cores, Accelerators, and More*. [Online]. Available: https://github.com/ucb-bar/chipyard

[47] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic, "The RISC-V instruction set manual, volume I: User-level ISA, version 2.0," Dept. EECS, Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-54, May 2014. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-54.html

[48] SiFive. (Aug. 2019). *SiFive TileLink Specication*. [Online]. Available: https://www.sifive.com/documentation/tilelink/tilelink-spec/

[49] SiFive. (2018). *Freedom U540-C000 Bootloader Code*. [Online]. Available: https://github.com/sifive/freedom-u540-c000-bootloader

[50] J. Strömbergson and O. Kindgren. (2021). *Verilog Implementation of the Symmetric Block Cipher AES (NIST FIPS 197)*. [Online]. Available: https://github.com/secworks/aes

[51] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A testing methodology for side-channel resistance validation," in *Proc. NIST Non-Invasive Attack Test. Workshop*, vol. 7, 2011, pp. 115–136.

[52] J. Cooper, G. Goodwill, J. Jaffe, G. Kenworthy, and P. Rohatgi, "Test vector leakage assessment (TVLA) methodology in practice," in *Proc. Int. Cryptograph. Module Conf.*, vol. 20, 2013.

[53] J. Yang, J. Han, F. Dai, W. Wang, and X. Zeng, "A power analysis attack resistant multicore platform with effective randomization techniques," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 6, pp. 1423–1434, Jun. 2020.

[54] B. L. Welch, "The generalization of 'student's' problem when several different population varlances are involved," *Biometrika*, vol. 34, nos. 1–2, pp. 28–35, 1947.

[55] Xilinx. (Jul. 2018). *7 Series FPGAs Clocking Resources User Guide UG472 (V1.14)*. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf

[56] (Mar. 2021). *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics DS182 (V2.19)*. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds182_Kintex_7_Data_Sheet.pdf

[57] J. Tatsukawa. (Aug. 2019). *MMCM and PLL Dynamic Reconfiguration XAPP888 (V1.8)*. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp888_7Series_DynamicRecon.pdf

[58] (2018). *ASCAD Database*. [Online]. Available: https://github.com/ANSSI-FR/ASCAD

[59] (2010). *Database of DPA contest V2*. [Online]. Available: http://www.dpacontest.org/v2/download.php

[60] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Berlin, Germany: Springer, 2007.

[61] A. Alipour, A. Papadimitriou, V. Beroulle, E. Aerabi, and D. Hely, "On the performance of non-profiled differential deep learning attacks against an AES encryption algorithm protected using a correlated noise generation based hiding countermeasure," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 614–617.

[62] H. Liu, X. Wang, and A. Kadir, "Constructing chaos-based hash function via parallel impulse perturbation," *Soft Comput.*, vol. 25, pp. 11077–11086, May 2021.

[63] M. Alawida, A. Samsudin, N. Alajarmeh, J. S. Teh, M. Ahmad, and W. H. Alshoura, "A novel hash function based on a chaotic sponge and DNA sequence," *IEEE Access*, vol. 9, pp. 17882–17897, 2021.

[64] S. Dong, Y. Wang, X. Xin, and X. Tong, "A chaos-based true random number generator based on OTA sharing and non-flipped folded Bernoulli mapping for high-precision ADC calibration," *Microelectron. J.*, vol. 116, Oct. 2021, Art. no. 105259.

[65] A. C. Özçelık and Z. G. C. Taskiran, "Chaotic oscillator based true random number generator," in *Proc. 29th Signal Process. Commun. Appl. Conf. (SIU)*, Jun. 2021, pp. 1–4.

[66] I. S. Doubla, Z. T. Njitacke, S. Ekonde, N. Tsafack, J. D. D. Nkapkop, and J. Kengne, "Multistability and circuit implementation of Tabu learning two-neuron model: Application to secure biomedical images in IoMT," *Neural Comput. Appl.*, vol. 33, pp. 14945–14973, Jun. 2021.

[67] Z. T. Njitacke, M. E. Sone, T. F. Fozin, N. Tsafack, G. D. Leutcho, and C. T. Tchapga, "Control of multistability with selection of chaotic attractor: Application to image encryption," *Eur. Phys. J. Special Topics*, vol. 230, pp. 1–16, May 2021.

[68] Z. T. Njitacke, S. D. Isaac, T. Nestor, and J. Kengne, "Window of multistability and its control in a simple 3D Hopfield neural network: Application to biomedical image encryption," *Neural Comput. Appl.*, vol. 33, no. 12, pp. 6733–6752, Jun. 2021.

[69] M. S. Açikkapi, F. Özkaynak, and A. B. Özer, "Side-channel analysis of chaos-based substitution box structures," *IEEE Access*, vol. 7, pp. 79030–79043, 2019.

[70] S. Zhang, Y. Luo, L. Cao, and J. Liu, "Cryptanalysis of a chaos-based block cryptosystem using multiple samples correlation power analysis," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 892–898.

[71] A. A. El-Moursy, A. M. Darya, A. S. Elwakil, A. Jha, and S. Majzoub, "Chaotic clock driven cryptographic chip: Towards a DPA resistant AES processor," *IEEE Trans. Emerg. Topics Comput.*, early access, Dec. 21, 2020, doi: 10.1109/TETC.2020.3045802.

BA-ANH DAO (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and telecommunications and the M.S. degree in microelectronics from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2014 and 2019, respectively. He is currently pursuing the Ph.D. degree in information and network engineering with The University of Electro-Communications (UEC), Tokyo, Japan. He is also a Research Assistant with the Academy of Cryptography Techniques (ACT), Hanoi.

TRONG-THUC HOANG (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and telecommunications and the M.S. degree in microelectronics from the University of Science, Vietnam National University, Ho Chi Minh City, Vietnam, in 2012 and 2017, respectively. He is currently pursuing the Ph.D. degree in information and network engineering with The University of Electro-Communications (UEC), Tokyo, Japan. He is also a Research Assistant with the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo.

ANH-TIEN LE (Graduate Student Member, IEEE) received the M.S. degree in information systems from the Hanoi University of Science and Technology, Vietnam, in 2019. He is currently pursuing the Ph.D. degree in information and network engineering with The University of Electro-Communications (UEC), Tokyo, Japan. He is also a Lecturer with the Academy of Cryptography Techniques (ACT), Hanoi, Vietnam.

AKIRA TSUKAMOTO received the M.S. degree in computer science from Columbia University, New York, NY, USA. He currently works at the National Institute of Advanced Industrial Science and Technology (AIST). He has worked on products based on Cell/B.E. and ARM. His main research interests include software engineering on networks, operating systems, and system security, and he is enthusiastic regarding any kind of technical development.

KUNIYASU SUZAKI (Member, IEEE) received the B.E. and M.E. degrees in computer science from the Tokyo University of Agriculture and Technology and the Ph.D. degree in computer science from The University of Tokyo, Tokyo, Japan. He is currently a Senior Researcher with the National Institute of Advanced Industrial Science and Technology (AIST) and a Researcher with the Technology Research Association of Secure IoT Edge Applications Based on the RISC-V Open Architecture (TRASIO). His research interests include security on CPUs, operating systems, and hypervisors.

CONG-KHA PHAM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Sophia University, Tokyo, Japan. He is currently a Professor with the Department of Computer and Network Engineering, The University of Electro-Communications (UEC), Tokyo. His research interests include the design of analog and digital systems using integrated circuits.