RESEARCH ON HARDWARE-BASED HIDING COUNTERMEASURES AGAINST POWER ANALYSIS ATTACKS

BA-ANH DAO

A dissertation submitted for the degree of

Doctor of Engineering

Graduate School of Informatics and Engineering
Department of Computer and Network Engineering
The University of Electro-Communications

March 2022

Research On Hardware-based
Hiding Countermeasures Against Power Analysis Attacks

APPROVED

Prof. Cong-Kha PHAM, Chairman

Prof. Koichiro ISHIBASHI

Prof. Akashi SATOH

Prof. Kazuo SAKIYAMA

Assoc. Prof. Takeshi SUGAWARA

Date approved by Chairman: ____________________

This page intentionally left blank

I would like to dedicate this dissertation to my entire family and friends,
who have been constant sources of support and encouragement during the challenges of
graduate school and life.

This page intentionally left blank

Acknowledgements

First and foremost, I would like to express my heartfelt appreciation to my supervisors, Prof. Cong-Kha Pham and Prof. Koichiro Ishibashi, for their unwavering support and guidance throughout my PhD studies and research at the University of Electro-Communications. Without their patience, motivation, encouragement, and knowledge, I would never have been able to finish my studies.

I would also like to thank all the committee members, including Prof. Akashi Satoh, Prof. Kazuo Sakiyama, and Assoc. Prof. Takeshi Sugawara, for their valuable comments and recommendations on my dissertation.

Aside from my advisors, I'd like to thank all of the members of Pham-Lab at the University of Electro-Communications for all the discussions, the sleepless nights we spent working together before deadlines, and all of the fun we have had over the last three years.

My gratitude also goes to the Vietnam Government Information Security Commission for allowing me to work as their research assistant and obtain my very first research experience, as well as for providing me with the scholarship that allows me to pursue my PhD at the University of Electro-Communications. Besides, I also thank all the VGISC students at UEC for their generous assistance, daily life advice, and sharing while we are thousands of miles away from home.

Last but not least, I would like to thank my whole family in Vietnam, especially my parents and my little sister, for their unconditional love, support, and belief in me throughout my life.

This page intentionally left blank

This page intentionally left blank

論文概要

近年、電力解析攻撃は、暗号化デバイスのセキュリティに対する深刻な脅威として浮上している。これらの攻撃は、対応する暗号化アルゴリズムの数学的弱点の分析に焦点を合わせるのではなく、暗号化の実装によって引き起こされる意図しない情報漏えいを悪用するため、サイドチャネル攻撃の一種として分類される。意図しないリークは、暗号化機能で機密性の高い中間値を実行する際の、対象の暗号化デバイスの電力消費または電磁放射である。この漏洩した情報を取得して分析することにより、攻撃者は対応する暗号化プロセスで使用される秘密鍵を取得できる。

電力解析攻撃に対する多くの対策が提案されている。これらのアプローチに基づく対抗策は、隠蔽対抗策とマスキング対抗策の2つのグループに分類できる。マスキング対策では、暗号化アルゴリズムが変更され、中間値がランダムにマスクされるため、暗号化デバイスによって処理される実際の中間値は予測できなくなる。一方、隠蔽対策では、暗号化デバイスの消費電力特性を変更することにより、意図しないリークの中間値への依存性を大幅に低減させる。

マスキング対策は複雑であり、多くの場合、暗号化アルゴリズムの計算の複雑さを大幅に増加させ、スループットとパフォーマンスを低下させ、実装リソース使用率を増加させてしまう。さらに、ディープラーニングを利用した最先端の電力分析攻撃は、使用されているマスクに関する事前の知識がなくても、マスクされた暗号化デバイスを破る可能性があることも示した。

一方、隠蔽対策は暗号化アルゴリズムに影響を与えず、より簡単で、より適している。隠蔽対策は、同じ暗号化デバイスに一度に統合された複数の暗号化アルゴリズムをカバーすることもでき、マスキング対策よりもディープラーニングベースの電力分析攻撃に対する保護が優れていることが実験的に証明されてる。これにより、ハードウェアの使用率、パフォーマンス、および隠蔽対策の複雑さに関するオーバーヘッドは、マスキング対策のオーバーヘッドよりも小さいが、これらのオーバーヘッドは依然としてかなりのレベルにある。

本論文は、隠蔽対策のさらなる開発に焦点を当てており、2つの新しいハードウェアベースの隠蔽対策を提案する。提案利点は、電力分析攻撃に対する適切な有効性、可用性、および単純さを有する。最初の提案は、一般的なFD-SOI製造技術のバックゲートバイアス技術の活用に関する。バックゲートバイアス技術は、製造されたデバイスの性能と消費電力を動的に制御するために広く使用されている。提案対抗策は、追加のトレードオフなしに電力分析攻撃に抵抗するためにこの技術を利用することを提案する。2番目の提案は、ランダム動的周波数スケー

リング対策であり、暗号化デバイスの動作周波数を動的に再構成する機能を限界まで押し上げて、多数の個別周波数を実現する。

提案対抗策は、提案対抗策を適用する場合としない場合で、FPGA と ASIC の両方に実装された実際の暗号化デバイスに対して実際の電力分析攻撃を実行し評価する。評価において、最先端の電力分析攻撃の Vector Leakage Assessment（TVLA リークテスト）、Correlation Power Analysis（CPA）攻撃や Deep Learning-based Side チャネル攻撃（DL-SCA）などを実施した。実験結果により、提案対策は、特にハードウェアリソースの使用率の点で、他の隠蔽対策よりも優れていることが分かった。

Abstract

In recent years, Power Analysis attacks have emerged as a severe threat to the security of cryptographic devices. These attacks are categorized as a type of side-channel attack since they rather exploit the unintended information leakage caused by the cryptographic implementations than focus on analyzing the corresponding cryptographic algorithms' mathematical weaknesses. The unintended leakages are power consumption or electromagnetic radiation of the targeted cryptographic device when it executes sensitive intermediate values in cryptographic functions. By acquiring and analyzing this leaked information, adversaries can retrieve the secret key used in corresponding cryptographic processes.

Numerous countermeasures against Power Analysis attacks have been proposed. Based on their approaches, these countermeasures can be classified into two groups, which are hiding countermeasures and masking countermeasures. In masking countermeasures, the cryptographic algorithms are modified to randomly mask the intermediate values so that the actual intermediate values processed by the cryptographic device is unpredictable. Meanwhile, in hiding countermeasures, the dependency of the unintended leakage on the intermediate values is significantly reduced by altering the cryptographic device's power consumption characteristics. All masking countermeasures are complex and often significantly increases the cryptographic algorithm's computational complexity, reduces throughput and performance, and increases resource utilization of its implementation.

Furthermore, the state-of-the-art power analysis attack utilizing Deep Learning also demonstrated that it could break the masked cryptographic devices without any prior knowledge about the used masks. On the other hand, the hiding countermeasures leave the cryptographic algorithm untouched. Therefore, they are more straightforward and more suitable in practice. The hiding countermeasures also can cover multiple cryptographic algorithms integrated into the same cryptographic device at once. They are also experimentally proved to provide better protection against Deep Learning-based Power Analysis attacks than the masking countermeasures. Hence, the overheads in terms of hardware utilization, performance and complexity of hiding countermeasures are often smaller than that of masking countermeasures, but these overheads are still at significant levels.

This dissertation, therefore, focuses on further developing the hiding countermeasures. Two novel hardware-based hiding countermeasures are proposed in this dissertation. The proposed countermeasures' advantages include their decent effectiveness against power analysis attacks, availability, and simplicity. The first proposed hiding countermeasures is the exploitation of the Back-gate biasing technique of popular FD-SOI fabrication technology. The back-gate biasing technique has been widely used to control fabricated devices' performance and power consumption dynamically. The proposed countermeasure suggests utilizing this technique to resist power analysis attacks without any additional trade-off. The second proposal is the Random Dynamic Frequency Scaling countermeasure, where the ability to reconfigure the operating frequency of the cryptographic devices dynamically is pushed to its limit to achieve an extremely high number of distinct frequencies. All proposed countermeasures are evaluated by conducting practical power analysis attacks on actual cryptographic devices, implemented in both FPGAs and ASICs, with and without applying the proposed countermeasures. Various popular, state-of-the-art power analysis attacks are used in the evaluation of two proposed countermeasures, including the Test Vector Leakage Assessment (TVLA leakage test), the Correlation Power Analysis (CPA) attacks and the Deep Learning-based Side-Channel Attacks (DL-SCA). Based on the experiment results, these proposed countermeasures outperform other hiding countermeasures, especially in terms of hardware resources utilization.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AE	Auto Encoder
AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
BADDL	Body-biased Adiabatic Dynamic Differential Logic
BOX	Buried Oxide
CLINT	Core Local Interrupts
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CP	Charge Pump
CPA	Correlation Power Analysis
CPU	Centre Processing Unit
DDR	Double Data Rate
DFR	Dynamic Frequency Randomization
DLSCA	Deep Learning based Side Channel Attack
DPA	Differential Power Analysis
DRP	Dynamic Reconfiguration Port
DTLS	Datagram Transport Layer Security
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FDSOI	Fully Depleted Silicon on Insulator
FPGA	Field Programmable Gate Array
FPR	Frequency and Phase Randomiztion
GIDL	Gate Induced Drain Leakage

GPIO	General Purpose Input-Output
HD	Hamming Distance
HW	Hamming Weight
IoT	Internet of Things
ISA	Instruction Set Architecture
LF	Loop Filter
LSTM	Long and Short Term Memory
MCU	Micro-Controller Unit
MLP	Multilayer Perceptron
MMCM	Mixed Mode Clock Managers
MMIO	Memory Mapped Input-Output
NVDLA	NVIDIA Deep Learning Accelerator
PCB	Printed Circuit Board
PFD	Phase Frequency Detector
PGE	Partital Guess Entropy
PLIC	Platform Level Interrupt Controller
PLL	Phase Lock Loop
POI	Point of Interest
PRNG	Pseudo Random Number Generator
PSMC	Power State Monitoring Control
RDBB	Random Dynamic Back-gate Bias
RDFS	Random Dynamic Frequency Scaling
RDVFS	Random Dynamic Voltage Frequency Scaling
RDVS	Random Dynamic Voltage Scaling
RFTC	Runtime Frequency Tuning Countermeasure

RIO	Random Insertion of Operations
RISC-V	Reduced Instruction Set Computer Five
ROM	Read Only Memory
RSA	Rivest–Shamir–Adleman
RTS	Random Task Scheduling
SCA	Side Channel Attacks
SHA	Secure Hash Algorithm
SoC	System on Chip
SOTB	Silicon on Thin BOX
SPA	Simple Power Analysis
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
STI	Shallow Trench Isolation
TEE	Trusted Execution Environment
TVLA	Test Vector Leakage Assessment
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VCO	Voltage Control Oscillator
VLC	Visible Light Communication
VLSI	Very Large Scaled Integrated
WDDL	Wave Dynamic Differential Logic

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Power Analysis Attacks

To ensure data confidentiality, integrity, and validity, cryptographic algorithms are increasingly being used in modern times. Communications, computer networks, and computer security in general have all benefited from the widespread usage of cryptographic algorithms. These algorithms make use of a cryptographic key to turn conventional information, known as plaintext, into an unintelligible form, known as ciphertext. Modern cryptographic algorithms are frequently made available for free online. Their reliability is only dependent on the confidentiality of the cryptographic key. Typically, breaking a cryptographic algorithm means determining the secret key that enables attackers to reverse the ciphertext and retrieve the original data. The study of analyzing cryptographic systems to break them is called cryptanalysis. Since cryptographic algorithms are often designed using intractable mathematical problems, every detailed step of a modern cryptographic algorithm can be made publicly available. In practice, a cryptographic algorithm is deemed secure if no attack scheme can be developed to break it within an acceptable amount of computing power and processing time. Modern cryptographic algorithms are also constructed in such a way that the computational cost of breaking them increases exponentially with the length of the secret key. As a result, the number of secret key bits used in a cryptographic algorithm is a significant factor in determining the security of the algorithm. Various cryptographic algorithms, such as Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA), and CHACHA20, are now popular and believed to be computationally secure.

Previously, mathematical analysis techniques have been extensively used in cryptanalysis. These techniques are limited to exploiting weaknesses in the cryptography algorithms themselves. Numerous advanced cryptographic algorithms have been proposed and demonstrated to be resistant to ordinary mathematical cryptanalysis through

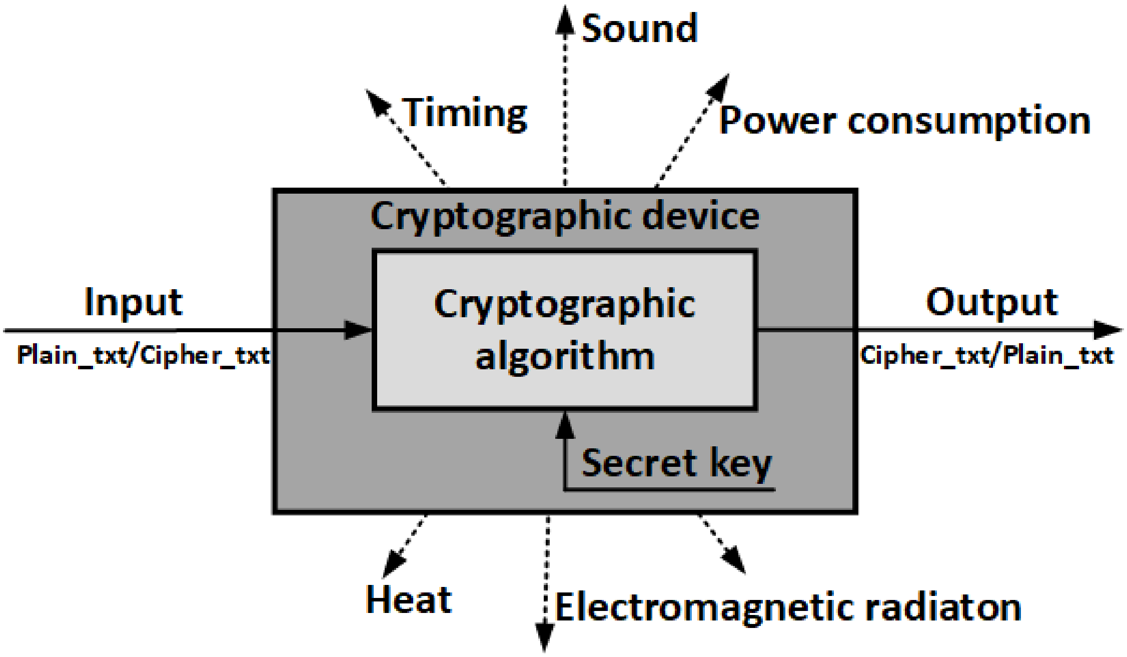


Figure 1.1: A cryptographic device and side-channel leakages.

substantial academic research. However, cryptographic algorithms cannot be used in practice unless they are implemented in physical devices. Cryptographic implementation may take the form of software implementations in microprocessors, microcontrollers, or smart cards. Additionally, it could be hardware implementation in the form of Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), or hardware accelerators in System on Chip (SoC). Thus, in addition to mathematical research of cryptographic algorithms, cryptanalysis includes the study of side-channel attacks that exploit vulnerabilities in cryptographic implementations rather than in the cryptographic algorithms themselves. Timing information, power consumption, electromagnetic radiation, and even emitted sound can be used to extract sensitive information in side-channel attacks. Figure 1.1 illustrates a typical cryptographic device, which implementing an cryptographic algorithm. When the device operates in encryption mode, it takes plaintext as input and produces ciphertext as output. When the device operates in decryption mode, it takes ciphertext as input and produces plaintext as output. During the encryption/decryption processes, the cryptographic device always emits side-channel leakages in forms of timing information, sounds, power consumption, heat, or electromagnetic radiation.

Power analysis attacks are a subset of side-channel attacks. The attacker uses leaked information about a cryptographic device's power consumption or electromagnetic radiation to recover secret keys needed in cryptographic operations. In 1999, an early version of the power analysis attack was developed by Kocher *et al.* [1]. They demonstrated that by directly observing a power trace of the cryptographic device during its operation, the Simple Power Analysis (SPA) attack could tell which kind of cryptographic algorithm is being executed. Additionally, Differential Power Analysis (DPA) was also introduced. Based on the general knowledge about the cryptographic device, the attackers can compute all possible hypothetical power consumption values by using a simple power consumption model called binary-bit model. The hypothetical values is subsequently compared with measured power traces using the difference-of-means statistical function to determine the most likely secret key used in the cryptographic device. The methodology of DPA attacks is developed based on the fact that the instantaneous power usage is highly dependent on the intermediate data processed. Numerous authors have proposed different statistical functions and power models to increase the DPA attack's efficiency. For examples, these proposals are using the Hamming-Weight power model [2], combining usages of the Hamming-Distance power model and Pearson's correlation coefficient [3], using the maximum-likelihood test [4] or the least-squares test as alternative statistical functions [5]. Methods for pre-processing the measured power traces have also been introduced to lower the computing complexity of DPA attacks while boosting their success rate. Several pre-processing methods that can be named are using marker locations [6], using wavelet transform [7], the trace integration method [8], and the principal component analysis [9]. As a result, DPA attack has gone mainstream as a simple-yet-effective approach for compromising the security of a cryptographic device, even thought the adversaries have only a cursory understanding of the targeted device's power consumption characteristics. For instance, an attacker might use an open source SCA toolchain to successfully compromise the security of an conventional AES-128 software implementation on a general-purpose microcontroller Power traces measured from only 50 encryptions are required to get the secret encryption key [10]. Other types of cryptographic devices, including ASICs, FPGAs, and smart-cards, are also vulnerable to Differential Power Analysis attacks [11, 12, 13]. Recent research also has utilizing the power of Machine Learning and Deep Learning novel power analysis attack techniques [14, 18, 19, 20]. When attacking cryptographic device with desynchronized countermeasures, these methods were demonstrated to surpass conventional DPA attacks in the rate of success.

Power Analysis attacks can be categorized 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. For example, measuring equipment such as oscilloscopes can be used to measure the power consumption of the target device. The measured data are saved as power traces. Meanwhile, the attackers only have general knowledge about the power consumption characteristics of the targeted device and can only construct hypothetical power models to use in the statistical analysis process. Hence, they use a non-profiled analysis to extract the secret key from the measured power traces. Figure 1.2 shows the common process of non-profiled attacks. The Correlation Power Analysis (CPA) attack, the Differential Power Analysis (DPA) attack, and the Simple Power Analysis (SPA) attack are several popular instances of non-profiled PA attacks. Both of DPA and SPA attack are introduced in 1999 by Kocher *et al.* [1]. 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 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 2004, Brier *et al.* proposed using Pearson's correlation coefficient as a statistical parameter to improve the efficiency of the power analysis attacks [3]. 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. The common process of profiled power analysis attacks is described in Figure 1.3. The attackers measure the power traces of an profiling device during its cryptographic operations with various different keys. These profiling traces are then used to build the leakage profile, which is the power consumption characteristics of the profiling device. Since

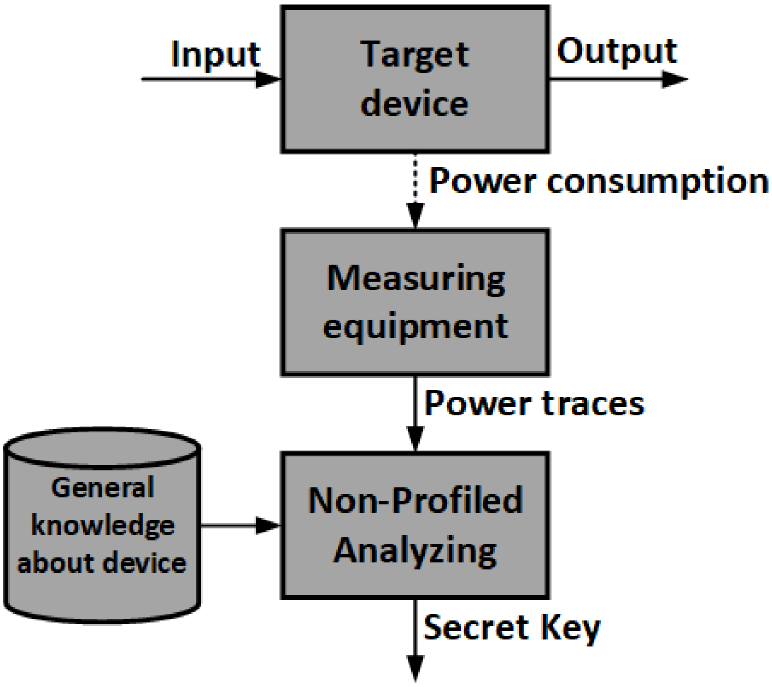


Figure 1.2: Common process of non-profiled power analysis attacks.

the profiling device and the target device are identical, the constructed leakage profile can also be used to describe the power consumption characteristic of the target device. The attackers can now measure the power traces from the target device and analyze these traces with the leakage profile to extract the secret key. Unfortunately, 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 Linear Regression Analysis [22], and Template Attacks [21]. Deep Learning (DL) and Machine Learning (ML) techniques are also used in profiled power analysis attacks because of the strong similarity between profiled attacks and supervised learning. Some DL-based profiled power analysis attacks were presented by Benadjila *et al.* in 2018 [17], by Cagli *et al.* in 2017 [16], and by Maghrebi *et al.* in 2016 [15]. Moreover, in 2019, Timon *et al.* demonstrated that DL-based attacks could also be applied in the non-profiled scenarios [20]. Meanwhile, several ML-based profiled power analysis attacks are presented by Hou *et al.* in 2017 [24], by Duan *et al.* in 2015 [18], and by Chakraborty emphet al. [23].

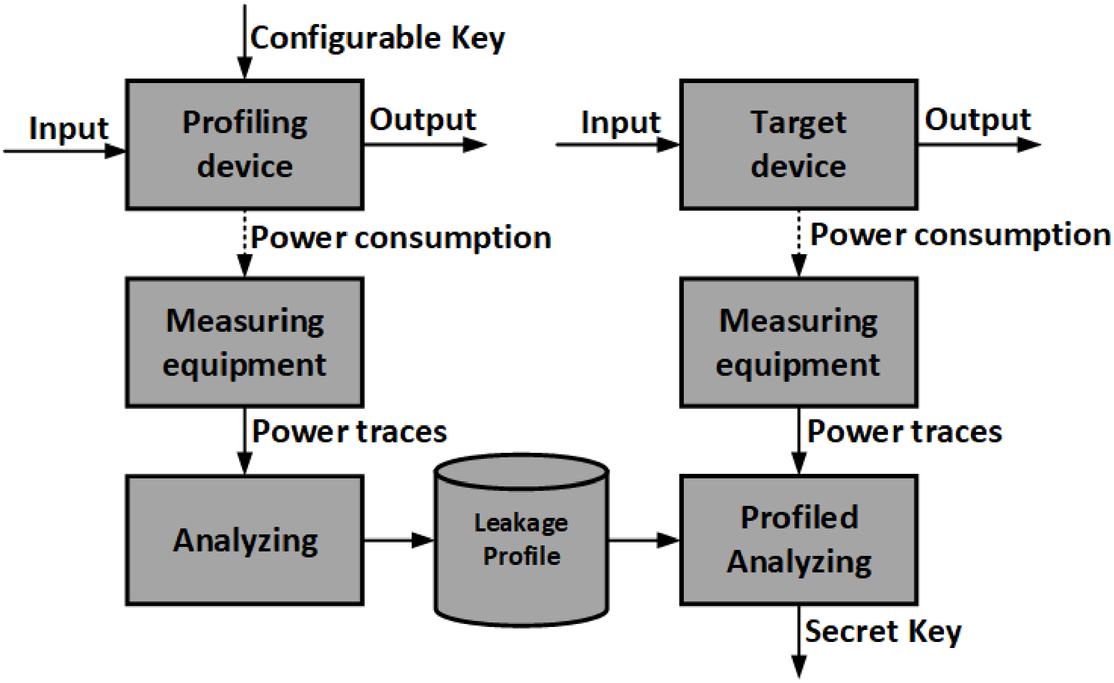


Figure 1.3: Common process of profiled power analysis attacks.

1.2 Countermeasures Against Power Analysis Attacks

1.2.1 Masking countermeasures

Power analysis attacks are effective because the power consumption of cryptographic devices is dependent on the intermediate values of the cryptographic algorithms that are being run. As a result, the fundamental idea behind countermeasures against power analysis attacks is the elimination of this dependency. Numerous countermeasures have been developed in response to power analysis attacks. They are categorized into two categories: masking countermeasures and hiding countermeasures. The masking countermeasures are developed based on the premise that side-channel leakage is produced by the usage of sensitive intermediate values in the cryptographic algorithms, hence the masking countermeasures attempt to disguise the intermediate values using random values. Several approaches have been introduced as masking countermeasures, including secret sharing, arithmetic masking, and Boolean masking [25, 26], as well as the threshold cryptography [27, 28]. The primary advantage of using masking countermeasures is that the device's power consumption characteristics do not need to be altered. Power consumption may continue to be data-dependent. Since the cryptographic device processes only randomized intermediate values, conventional power analysis attacks are prevented. Additionally, masking countermeasures may be deployed easily at the software level of cryptographic systems. This advantage allows front-end users to improve the applied masking countermeasures at anytime. However, the downside of masking countermeasures is that they require modifications to cryptographic algorithms. This requirement substantially increases the computational complexity of the cryptographic algorithm, decreases its throughput and performance, and escalates the algorithm's implementation's resource usage. Furthermore, a particular masking scheme can be used solely to the algorithm for which it was created. When several cryptographic algorithms are integrated into a single device, the penalties associated with the implementation of each masked method are combined together. Higher order power analysis attacks are also reported to be able to defeat lower order masking countermeasures. Additionally, [20, 17] demonstrated that Deep-Learning-based power analysis attacks may defeat masked cryptographic devices without prior knowledge of the masking scheme.

1.2.2 Hiding countermeasures

Apart from masking countermeasures, another class of countermeasures is known as hiding countermeasures. The fundamental concept of hiding countermeasures is to break the dependency of power consumption on the intermediate data being processed. These countermeasures modify the power consumption characteristics of cryptographic device in order to drastically minimize its reliance on intermediate values. This objective can be accomplished in two ways. The first method is to build the cryptographic device in such a manner that it consumes roughly the same amount of power when performing every operation. The second method is to make the power consumption of the cryptographic device is more or less random. Numerous hiding countermeasures have been introduced, including using DPA-resistant logic styles [33, 34, 35], randomly changing the supply voltage and clock frequency of circuits [32], adding a voltage regulator [31], flattening the current consumption [30], and increasing the noise level [29].

In practice, hiding countermeasures is preferable since the cryptographic algorithms remain untouched and free of unexpected mathematical vulnerabilities. Cryptographic devices protected by hiding countermeasures process the same intermediate values as unprotected devices. Since the device's power consumption characteristics are modified, hiding countermeasures are frequently performed at the physical level of the device. By employing hiding countermeasures, it is possible to cover numerous cryptographic algorithms simultaneously. As a result, overheads in terms of area, power consumption, and performance are frequently reduced to a minimum. Additionally, Alipour *et al.* evaluated a noise-generation-based hiding countermeasure against non-profiled Deep Learning based SCA, [36]. Alipour *et al.* demonstrated that a hiding countermeasure based on noise generation would be more effective than a masking countermeasure in protecting against the state-of-the-art machine learning and deep learning-based power analysis attacks. The disadvantage of hiding countermeasures is that in practice, the data dependency cannot be entirely eliminated. Thus, employing a hiding countermeasure only increases the amount of power consumption traces that an attacker must analyze in order to effectively extract the correct secret key. Hiding countermeasure cannot completely prevent power analysis attacks.

1.3 Motivation And Key Contributions

Despite the fact that numerous hiding and masking countermeasures have been developed throughout the years. Attacks based on power analysis are still not entirely avoided. Attackers have always devised new attack methods or enhanced old ones in order to circumvent the cryptographic system's security. As a result, there has never been a greater need for novel countermeasures against power analysis attacks. However, hiding countermeasures are much less popular than masking countermeasures. The primary reason for this is because integrating hiding countermeasures is frequently done at the very beginning of the cryptographic system's design and fabrication processes. These processes frequently necessitate the use of proprietary tools and designs that are not publicly available. Fortunately, in recent years, the trend of sharing open-source tools and designs for developing customized hardware has begun to gain momentum. The RISC-V Foundation has announced the release of the RISC-V open Instruction Set Architecture (ISA) under open-source licensing. Apart from the instruction set architecture, there are several open-source RISC-V hardware designs available. It has become a fascinating topic for a large number of SoC designers in academia and industry, as it eliminates the majority of the constraints associated with working with proprietary design tools and SoC components.

Designers may effortlessly customize a RISC-V SoC to meet their specific requirements. Within a few years, the RISC-V SoC has gained widespread adoption and is being used in a wide variety of applications. For example, Feng *et al.* incorporated the NVIDIA Deep Learning Accelerator (NVDLA) onto a RISC-V SoC to run the LeNet-5 and accelerate the process of identifying handwritten numerals up to 4,647 times [37]. Zhong *et al.* present a RISC-V SoC with integrated visible light communication (VLC) module for mobile payment applications [38]. Arnaud *et al.* offer a low-power, medical-grade RISC-V SoC for tissue stimulation. Among all of these many uses, security also continues to draw more attention. Therefore, this dissertation will make use of open-source RISC-V hardware designs as experimental subjects in order to further develop novel hiding countermeasures.

This dissertation proposed and evaluated two novel hiding countermeasures. The first proposed hiding countermeasure is to adjust the back-gate bias of cryptographic devices manufactured using the increasingly popular fully depleted silicon-on-insulator (FD-SOI) technology. Because of the elimination of the threshold voltage V_{th} fluctuation problem and improvement in controlling the gate-induced drain leakage (GIDL) current, FD-SOI technology is quickly displacing conventional planar Complementary Metal Oxide Semiconductor on bulk silicon technology. Additionally, the body-biasing technique offered by bulk CMOS technology starts to lose its effectiveness when the size of semiconductor transistors is reduced to less than 40 nm. Meanwhile, the FD-SOI processes supports back-gate biasing, which applies a voltage directly on the buried oxide layer. The back-gate biasing also modifies the electrostatic control characteristics of fabricated transistors and therefore modifies the transistor V_{th} in a similar manner with the bulk CMOS's body-biasing method. This feature enables fine-tuning of the FD-SOI semiconductor device to meet either high-performance or low-power requirements. Using back-gate bias in reverse region will eliminate leakage current at the expense of reducing device's performance. On the other hand, applying back-gate bias in forward region has the opposite effects. The device's transistors have more driving current, hence they perform better, but the current leakage is greater. Moreover, FD-SOI technology provides for a considerably larger range of back-gate bias than the old bulk CMOS technology [39], which offers a $\pm 300mV$ range of body bias. For a cryptographic device, high performance is frequently desired . Leakage current is viewed as undesirable design trade-off in these scenarios and must be optimized. This leakage current, on the other hand, contributes to the static proportion of total power consumption. The static power consumption is considered as the most significant noise source in power analysis attacks. As a result, the technique of back-gate biasing may be utilized to exploit leakage current and obfuscate the link between sensitive intermediate values and the instantaneous power consumption.

To validate the efficiency of the proposed countermeasure, realistic power analysis attacks on a target microcontroller are conducted, while the AES-128 algorithm is being processed. The target microcontroller is a 32-bit RISC-V microcontroller, fabricated on the 65 nm SOTB technology [40]. The results obtained from practical experiments demonstrate that using a fixed forward back-gate bias improves not only the performance of the targeted device but also its resistance to power analysis attacks, comparing to using no bias. Meanwhile, when a reverse back-gate bias is used to achieve low power consumption, vulnerability to power analysis attacks is remained the same. When a fixed forward back-gate bias is used in conjunction with a reduced supply voltage, the device's resistance to power analysis attacks becomes even more significant. Additionally, further improved resistance to power analysis attacks can be achieved by dynamically randomizing the back-gate bias of the targeted microcontroller. This proposed technique is named as Random Dynamic Back-gate Bias (RDBB).

It can be considered that the first proposed countermeasure aims to control the noise ratio in the power consumption trace, hence it affects the amplitude dimension of the power consumption trace. A second hiding countermeasure, which affects the time dimension of the power consumption trace, is also proposed in this dissertation. The second proposed hiding countermeasure is called Random Dynamic Frequency Scaling (RDFS), in which the cryptographic device's operating frequency is randomly scaled to cause random misalignment of the Point of Interest (POI)s. The RDFS countermeasure share the similar principle with various recent publications, including [42, 43, 44, 45]. However, the proposed method to generate different clock frequencies in this dissertation is improved in order to achieve the highest number of distinct frequencies. The similar principle used by other related work is using on-chip circuitry to generate a huge number of alternating clock signals from a constant off-chip oscillating source. These generated clock signals are used to drive cryptographic circuits. In these prior works, a different clock frequency is randomly chosen to operate the cryptographic circuit after each clock cycle, during the cryptographic operation. The corresponded authors view the number of distinct encryption completion times as a critical design parameter, where encryption completion time is defined as the total period of the multiple clock cycles required by a cryptographic circuit to perform an encryption operation. Their works aimed for the largest number of distinct completion times achievable. Encrypting completion times as a design parameter, in my opinion, is improper for the following reasons. To begin, in power analysis attacks, the attackers are solely concerned with the instantaneous power consumption of the targeted devices during the execution of the critical intermediate values. If these points of interest are well-aligned within the set of obtained power traces, power analysis attacks have a better chance of succeeding. For instances, consider a hardware implementation of AES-128 that could complete each encryption in ten clock cycles. When attacking an AES-128 implementation, intermediate values such as the outputs of S-Box substitution in the first round or the inputs of S-Box substitution in the final round may be exploited [3]. In each scenario, the POIs would be positioned in the first or tenth cycles of the power trace, respectively. Second, different completion times do not imply that the POIs in obtained power traces are misaligned. For instance, if the first round's S-Box substitution outputs are used as intermediate values, POIs may still be aligned even though the completion timings of each trace are different if the initial clock cycles of each trace are identical and the rest of the clock cycles are different. As a solution, the proposed RDFS countermeasure in this dissertation tries to avoid this issue by producing as many distinct frequencies as feasible and maintaining a fixed distinct frequency for each encryption/decryption. Only then can we be certain that the POIs in the set of observed power traces are heavily misaligned.

This dissertation has following key contributions:

1. This dissertation proposed two novel hiding countermeasures against power analysis attacks, one proposed countermeasure involves controlling the noise amplitude, while the other involves controlling the timing alignment of the POIs in the measured power consumption traces.

2. The first proposed countermeasure is exploiting the FD-SOI fabrication technology's well-known back-gate bias technique. Back-gate biasing has been extensively employed to dynamically adjust the power consumption and performance of fabricated devices. This study shows that the back-gate bias technique can also

be utilized to defend against power analysis attacks, without requiring any extra penalties.

3. The first proposed countermeasure's effectiveness is evaluated by performing various DPA attacks targeting a test microcontroller that is running the AES encryption. The target microcontroller is fabricated using the 65nm SOTB technology, which is a variant of the FD-SOI technology. Results obtained from practical experimets show that using reverse back-gate bias maintains the device's resistance against power analysis attacks. Meanwhile, applying a back-gate bias in forward region enhances the resistance against power analysis attacks. Additionally, if the back-gate bias is randomly changed in the forward bias region, the device's resistance is improved even further. Hence the first proposed countermeasure is named as Random Dynamic Back-gate Bias (RDBB).

4. Theoretical explanations are provided for the experimental outcomes of using the proposed RDBB countermeasure on the target device under different conditions. Suggestions of how to effectively use the proposed countermeasures is also provided. The RDBB countermeasure is aimed to be applied for any cryptographic device that fabricated using FD-SOI technology.

5. The second countermeasure against power analysis attacks, which dynamically adjust the operating frequency of cryptographic devices is also proposed in this dissertation. In comparison to previous studies, the proposed technique is capable of achieving the greatest number of different frequencies. The second countermeasure is named as Random Dynamic Frequency Scaling (RDFS) and aimed to be applied for complex SoCs integrated with cryptographic hardware accelerators.

6. Practical experiments are provided 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 cryptographic hardware accelerator are comparable with or even overshadowed by the unexploitable power caused by other SoC's components.

7. The efficiency of the proposed RDFS technique in enhancing resistance against power analysis attacks is evaluated using a variety of techniques, including the

TVLA leakage test, CPA attacks, and profiled DL-SCA attacks.

1.4 Dissertation Layout

This dissertation is divided into six chapters. Along with the introduction given in this chapter, the remaining content of this dissertation is organized in the following way:

- **Chapter 2** gives the literature review on recent hiding countermeasures. The relationship between this study and recent works is also provided.
- **Chapter 3** provides background knowledge related to various popular type of power analysis attacks, including the Simple Power Analysis, the Correlation Power Analysis, the Deep Learning based Side Channel Attack, the Test Vector Leakage Assessment. These power analysis attacks will be conducted to evaluate the novel hiding countermeasures proposed in this dissertation.
- **Chapter 4** describes in detail the first proposed hiding countermeasure, which exploiting the ability to adjust the back-gate bias of devices fabricated using FDSOI technology to enhance the resistance against power analysis attacks. The proposed hiding countermeasure in this chapter is called Random Dynamic Back-gate Bias. CPA attacks are conducted to assert the effectiveness of the RDBB countermeasure.
- **Chapter 5** proposes another novel hiding countermeasure named Random Dynamic Frequency Scaling. It discusses the detailed techniques used in the proposed RDFS countermeasure. The effectiveness evaluation is given in this chapter by performing CPA attacks, DLSCA attacks, TVLA tests on both protected and unprotected devices.
- **Chapter 6** summarizes the main results that this study has achieved. A list of current limitations and open topics for future researches are also provided.

Chapter 2

Literature Review

This chapter aims to review the motivation, methods, and accomplishments of recent related works on hiding countermeasures. The first section focuses on the other group of hiding countermeasures called amplitude-based hiding countermeasures. The common approach of amplitude-based hiding countermeasures is to directly modify the power consumption characteristics of the sensitive cryptographic operations. The second section focuses on the group of hiding countermeasures that try to perform the sensitive cryptographic operations at different instants of time during each encryption/decryption. This group of hiding countermeasures could be called as time-based hiding countermeasures. The relationship between the study in this dissertation and the literature review of hiding countermeasure is illustrated in Figure 2.1. In this dissertation, two novel hiding countermeasure are proposed and evaluated. One proposed countermeasure can be classified as a amplitude-based hiding countermeasure, while the other can be classified as a time-based hiding countermeasure.

2.1 Amplitude-based Hiding Countermeasures

Several amplitude-based hiding countermeasures were reviewed as listed below.

1. In [29], Benini *et al.* proposed a technique to protect cryptographic devices against differential power analysis attacks. The technique is a combination of power-reducing transformations and randomized clock gating. The proposed technique were applied to several implementations of the modular exponentiation modules, which are widely used in asymmetric cryptographic algorithms. Their experimental results showed that their proposed technique could achieve a significant amount of scrambling in the measured power traces, while maintaining and even reducing the level of power consumption.

2. In [46], Das *et al.* presented the Attenuated Signature Noise Injection (ASNI)

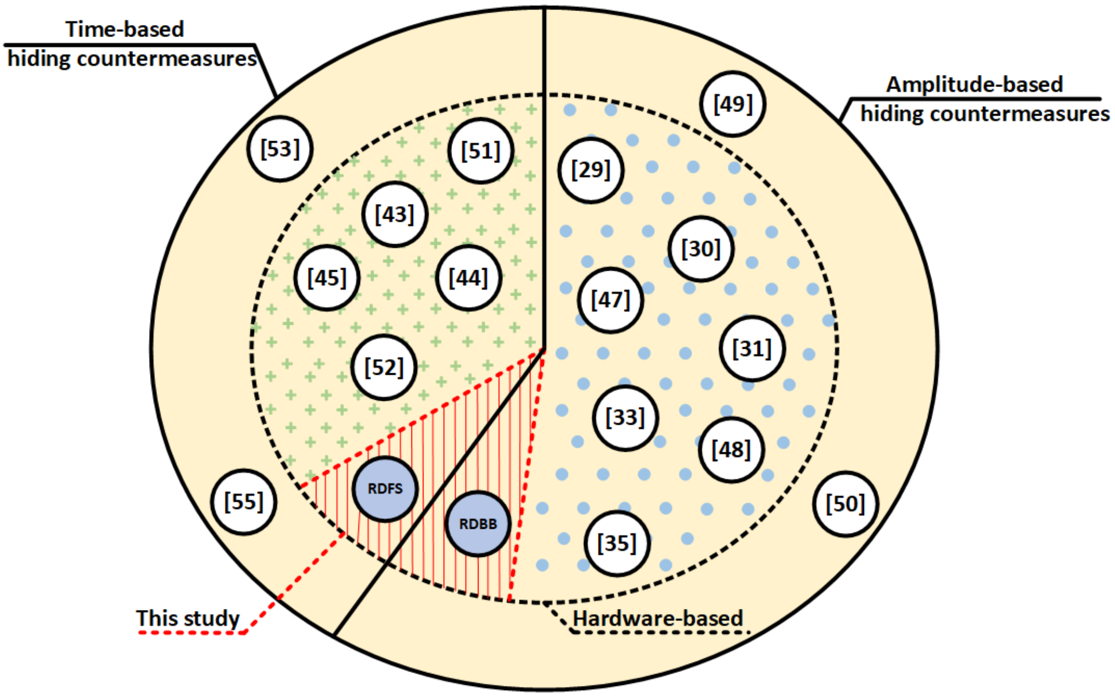


Figure 2.1: Literature review and this study.

method as a generic countermeasure against power analysis attacks. This countermeasure employs a shunt low-drop-out (LDO) regulator to reduce the cryptographic device's current consumption by more than 200 times. The LDO circuit is fabricated with 130nm CMOS technology and used to protect an AES-128 core running at 40 MHz. Das *et al.* conducted practical key extraction attacks on the protected AES-128 core and found that it can withstand CPA attacks on one million power traces without any penalty in performance of the cryptographic core. However, the trade-off was a 1.6× of area overhead.

3. In [30], Laohavaleeson *et al.* introduced an op-chip countermeasure to address the differential power analysis attacks. An current flattening circuit is integrated with the target cryptographic core. It senses the instantaneous current consumption of the cryptographic core and injects additional current to produce the constant total current consumption, thereby breaking the relation between the sensitive intermediate values being processed and the total power consumption of the target cryptographic device. The authors verified the effectiveness of their proposal by conducting DPA attacks on the simulated power traces obtained from the detailed layout level simulation of the current flattening circuit and an DES core. The showed that

it would be significantly harder to attack the protected core when using only 150 current consumption traces. However, when 2,000 traces were used, the DPA attacks found the correct secret key. Beside, by injecting extra current consumption, the power overhead was nearly double.

4. In [31], Kar *et al.* presented an power analysis countermeasure utilizing an on-die all-digital high frequency Inductive Voltage Regulator (IVR). The proposed IVR circuit includes a loop randomization module, a digital discontinuous conduction mode controller, and a digital reconfigurable proportional-integral-derivative (PID) controller. The authors conducted TVLA test and CPA attacks on an AES-128 core protected by the proposed IVR circuit. The CPA attacks using 100,000 traces failed to recover any byte of secret key. However, the TVLA test results on 10,000 power traces showed that the maximum t-score was around 8.5, which indicated that there were side-channel leakages existed in the measured power traces. Hence, the CPA attacks would have been succeed if more traces were used. Chip measurement results also showed that this proposed method only requires very low power, area, and performance overheads.

5. In [32] Yang *et al.* presented new designing approach to address the cryptographic system's vulnerabilities against power analysis attacks. Their proposed approach is named Dynamic Voltage and Frequency Switching (DVFS), since they dynamically switched the pair of voltage and frequency during the cryptographic operations. It is important to note that Yang *et al.* considered that a value of supply voltage is always paired with a value of operating frequency. In their experimental implementations, the supply voltage and operational frequency are both altered at the same time. The authors defined new metrics to evaluate the effectiveness of their proposed designing approach, which are the Signal Trace Entropy (STE), Energy Overhead (EO), and Time Overhead (TO). They claimed that by using their proposed countermeasure, the protected cryptographic system achieved high enough signal trace entropy to prevent power analysis attacks (SPA and DPA attacks), while the energy consumption is reduced by 27% and the time overhead is around 16%. However, no practical key extraction attack is conducted to verify this claim.

6. In [47], Baddam *et al.* pointed out a flaw of the DVFS designing technique pro-

posed in [32]. Attackers can utilize the SPA attacks and observe the glitches on the power line to detect the changes of operating frequency. Subsequently, attackers can exploit this information to conduct the more advance DPA attacks. Baddam *et al.* proposed an alternative countermeasure, where the operating frequency is kept unchanged and the supply voltage is randomly altered during cryptogpraphic operations. They provided DPA attack results on 10,000 simulated power traces of an AES S-box circuit. THe results showed that the DPA attacks were still able to reveal the secret key of the protected implementation. However, the number of traces required to reveal the secret key was increased by four times, while the correlation strenght of the correct key was lower by 10 times.

7. In [48], Singh *et al.* proposed to use an on-die all-digital security-aware low-drop-out (DLDO) regulator to enhance the resistance against power and electromagnetic analysis attacks. The proposed digital LDO regulator includes a randomized reference voltage generator, an all-digital clock modulation, a random switching noise injector, and a baseline LDO regulator using control loop induced perturbation. A test-chip contains the proposed DLDO circuit and an AES core is fabricated using 130nm CMOS technology. The authors conducted TVLA test and CPA attacks on the AES-128 core protected by the proposed DLDO regulator. The TVLA test results showed that using the proposed regulator significantly reduces the maximum peak of the t-score trace, from a figure of 258 down to only 13.1. However, since the maximum peak was greater than the ±4.5 limit, the TVLA results indicated that there were still leakages existed in the protected design. The CPA attacks using ten million measured power traces successfully revealed all 16 bytes of the secret key. Comparing to CPA attack results on the unprotected AES core, the number of required traces improved more than 3579 times. This is an impressive improvement in power analysis attack resistance. However, the proposed countermeasure also caused significant overheads, including 36.9% area overhead, 10.4% performance overhead, and 32% power overhead.

8. In [33], Tiri *et al.* introduced a dynamic and differential CMOS logic style to counter the power analysis attacks. The proposed CMOS logic style was designed to have a fixed amount of power consumption, regardless of the processing opera-

tions (or the input transition). The power consumption characteristics of the proposed CMOS logic style was compared with that of the conventional static complementary CMOS logic style. Comparison results showed that the proposed CMOS logic style helps to suppress the normalized energy variation by 116 times. However, the area and total power consumption of the implementations on the proposed CMOS logic style were always doubled when compared with similar implementations on conventional CMOS logic.

9. In [35], Andrews *et al.* presented a new logic style called body-biased adiabatic dynamic differential (BADDL) logic to prevent power analysis attacks on smart-cards. The proposed logic style aimed to make the protected device's power consumption characteristics independent of its logic switching activities. When cryptographic algorithms are implemented on the proposed logic style, all logical operations on different inputs would have a constant power consumption. Simulation results indicated that the all logic operations implemented on the proposed BADDL logic had the same average power consumption. Meanwhile, these conventional CMOS logic counterparts had different average power consumption for different logic operations.

10. In [49], Petrvalsky *et al.* proposed a constant-weight coding based software implementation of DPA countermeasure. The authors assigned a general constant-weight for each sensitive intermediate values used in the AES encrypting algorithm. By doing so, the actual values being proccessed in the cryptographic device always have the same Hamming weight. Hence, the instantaneous power consumption values corresponded to processing different sensitive intermediate values are balanced. Petrvalsky *et al.* implemented the proposed countermeasure on a ARM Cortex-M3 MCU. They conducts DPA attacks using Hamming weight model and the TVLA test on the target MCU to evaluate the effectiveness of their proposal. The experimental results showed that the constant-weight coding countermeasure can prevent the DPA attacks using the Hamming weight model. However, the TVLA test results with maximum t-score of 37 indicated that the side-channel leakage was still existed. More advanced power analysis attacks in the future might be able to defeat the constant-weight coding countermeasure.

11. [50], Kim *et al.* present the Power-Balancing software implementation. This countermeasure has a similar approach with that of the constant-weight coding countermeasure proposed by Petrvalsky *et al.* in [49]. However, Kim *et al.* improved the method since their approach does not use any Look-up tables. Their proposed method is also optimized for 32 bit microcontrollers, such as ARM, AVR32, *etc.* The experimental results showed that the Power-Balancing software implementation can thwart the CPA attacks using 10,000 analyzed traces.

2.2 Time-based Hiding Countermeasures

Several time-based hiding countermeasures were reviewed as listed below.

1. In [51], Yang *et al.* presented a power analysis resistant multi-core processor. This multi-core processor is equipped with four different countermeasures to create randomness distortion in both time and amplitude dimensions of the measured power traces. These four methods include a power state monitoring and control (PSMC) technique, a countermeasure called frequency and phase randomization (FPR), a countermeasure called random insertion of operations (RIO), and a countermeasure called random task scheduling (RTS). When all four countermeasures are activated, the proposed multi-core processor can endure CPA attacks and profiled power analysis attacks based on convolutional neural networks (CNNs) even with two millions analyzed power traces. However, when each technique is applied separately, the measurement to disclosure (MTD) only increased 45 times with PSMC countermeasure, 2 times with FPR countermeasure, 280 times with RIO countermeasures, and 7 times with RTS countermeasures. The RTS and RIO countermeasures are classified as time-based hiding countermeasures. In RTS countermeasure, the cryptographic operations are randomly switched among several processing cores. Meanwhile, in RIO countermeasure, extra dummy operation are randomly inserted into the encryption/decryption processes. Unfortunately, when several realignment techniques are applied, the MTD of RTS and RIO countermeasures significantly reduced, from 7 times down to 2.75 times, and from 280 times down to 3 times, respectively. On the other hand, Yang *et al.* demonstrated that the overheads caused by applying all four countermeasures are negligible, including 2.3% area overhead,

3.5% power overhead, and 4% performance overhead.

2. In [42], Guneysu *et al.* proposed several generic countermeasures against power analysis attacks that aimed for protecting FPGA devices. These countermeasures include Generating Gaussian Noise, Clock Randomization, Preventing Clock Frequency Manipulations, and Block Memory Content Scrambling. The Clock Randomization countermeasure can be considered as a time-based countermeasure. The authors utilized two Digital Clock Managers (DCMs) to produce eight phase-shifted clock signals at the same frequency, and then use nine clock buffers to select the driving signal for the target AES core from these eight clock signals. Practical CPA attack results indicated that around three million traces are required to recover the correct secret key. However, the driving clock signal of the AES core had a very low pulse rate. Hence, the protected AES core's throuhput was relatively low.

3. In [43], Jayasinghe *et al.* introduced a time-based countermeasure named Runtime Frequency Tuning Countermeasure (RFTC). This countermeasure use the dynamic reconfiguration feature of the Clock Managers provided in FPGAs to change the operating frequency of the target cryptographic core during its runtime. The authors carefully computed and selected the reconfiguration scheme so that they could generate thousands of frequencies and achieve more than 60,000 distinct encrypting completion times. The RFTC countermeasure is evaluated by conducting CPA attacks on preprocessed power traces with several preprocessing methods, including Dynamic Time Warping, Principle Component Analysis, and Fast Fourier Transform. The attack results showed that the protected AES core is safe from CPA attacks even when four million traces are analyzed.

4. In [44] Jayasinghe *et al.* proposed to use a countermeasure that similar with the one introduced in [43] to prevent the power analysis attacks on a soft processor system. They referred the proposed countermeasure as SCRIP. It shares the same approach with the RFTC countermeasure. A clock signal with random frequency is generated by using the Clock Manager to drive the protected cryptographic system. The proposed SCRIP countermeasure is applied to LowRISC, an open-source RISC-V processor. Practical CPA attack results showed that the SCRIP LowRISC is invulnerable with up to 300,000 analyzed traces. The TVLA test on 200,000 traces also

cannot detect any side channel leakage. Meanwhile, the SCRIP countermeasure causes 1.88× timing overhead and 1.04× hardware resources overhead.

5. In [45], Hettwer *et al.* proposed another generic hiding countermeasure based on dynamic clock frequency randomization. Their proposed countermeasure can generate a extremely unpredictable clock signal and achieve up to 20 million different encrypting completion times for the AES encryption operations. Similar to the other countermeasures proposed in [43, 44, 42], the Clock Managers on FPGA are employed to generate a wide range of different clock frequencies. In this work, Hettwer *et al.* demonstrated their proposal on a SoC, which includes a hard-processor core and a re-configurable FPGA part, since they want to make use of the ability to configure the FPGA part from the hard-processing core. The hard-processing core would generate the input clock signal for the FPGA part. The FPGA part subsequently create the extremely unpredictable clock signal from that input clock. Therefore, their proposed countermeasure can only be applied for the such hybrid systems of hard-processor and FPGA part. Practical experiment results are provided. It showed that the protected SoC can withstand CPA attacks, FFT-CPA attacks, Sliding window CPA attacks, and powerful DNN attacks using one million analyzed traces. TVLA test using five million traces also can not detect any side channel leakage in the measured traces.

6. In [52], El-Moursy *et al.* proposed using a chaos-based technique as a countermeasure against power analysis attacks for an AES processor. El-Moursy's countermeasure has a similar approach to the approaches of this work and other previously mentioned articles [42, 43, 44, 45]. In [52], 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 other related papers, ranging from a few hundred thousand up to several millions of power traces.

7. [53], Coron *et al.* presented a efficient algorithm for generation of random delays to improve device's resistance against power analysis attacks and fault injection attacks. The authors also introduced a method to quantifies the effectiveness of a random delay countermeasure. Practical CPA attack result demonstrated that their proposed countermeasure improved the measurement to disclosure metric by six times, while having the same performance penalty, compared with previously published work [54].

8. [55], Herbst *et al.* introduced an AES software implementation for 8-bit smart cards that is capable of resisting power analysis attacks. The proposed implementation is a combination of the first order masking countermeasure and the shuffling countermeasure. The shuffling countermeasure involved randomizing the sequence of cryptographic operations and deployed at software level. Herbst *et al.* provided practical attacks to demonstrated that their implementation is secure against several type of power analysis attacks, including simple power analysis attacks, template attacks, first order DPA attacks, and have proper resistance against high order DPA attacks.

Chapter 3

Preliminaries

3.1 Differential Power Analysis Attack

The most common form of power analysis attack is the Differential Power Analysis (DPA) attack. In 1999, the principle of DPA attacks is proposed and justified by Kocher *et al.* [1]. Using this sort of attack, attackers can obtain the secret key of a cryptographic device without having a thorough understanding of its architecture. The DPA attack is performed by collecting and analyzing the power traces from the target cryptographic device while cryptographic operations are being executed. Because of target device's physical properties, the sample values of power trace is frequently dependent on the data and operations currently being executed. To retrieve the device's secret key, attackers apply statistical methods to a set of observed power usage traces. It's worth noting that a single DPA attack only tries to reveal a subkey, which is a tiny portion of the secret key. To disclose the whole secret key, multiple DPA attacks must be conducted.

The most typical strategy for a DPA attack can be listed in several steps. Originally, the attacker need to be able to record either input or output of the target device. After that, the attacker selects an sensitive intermediate value of the target cryptographic algorithm. The selected intermediate value must be dependent on a known non-constant part of the recorded input or output, and on a subkey. In the AES algorithm, for example, there are numerous possibilities for selecting such sensitive values. Because the output of the S-Box substitution in the first AES's encryption/decryption round are dependent on the secret key and the plaintext/ciphertext input, attackers can pick them as intermediate values. Another possible option for selecting the intermediate values is the input of the S-Box substitution in the last AES's encryption/decryption round, since these inputs may be calculated directly from the secret key and the ciphertext.

Next, the power consumption of the targeted device while executing cryptographic operations on known, random data is recorded and stored as power traces. Additionally, either the input or output data associated with each encryption/decryption must also be logged. Because the same cryptographic operations are executed on same-size data inputs, all measured power traces must have the same sample length and aligned with each other in time axis. All measured power traces can be filled into a power trace data matrix of size $\boldsymbol{D}\times\boldsymbol{T}$, where $\boldsymbol{T}$ the trace length and $\boldsymbol{D}$ is the number of measured power traces.

The following step is to calculate the hypothetical intermediate values for each piece of known data and for each potential subkey value. This calculation generates a hypothetical intermediate value matrix of dimension $\boldsymbol{D}\times\boldsymbol{K}$, where $\boldsymbol{D}$ denotes the number of measured traces and $\boldsymbol{K}$ is the number of subkey hypotheses.

By using an appropriate power consumption model, each value in the hypothetical intermediate value matrix is translated to a hypothetical power consumption value. After mapping, the attacker has a hypothetical power consumption matrix with the dimensions $\boldsymbol{D}\times\boldsymbol{K}$. The attacker's fundamental knowledge about the targeted device must be used to determine or select a suitable power consumption model. The more information an attacker has about the target device, the more precise the model that may be created. The power consumption model, on the other hand, does not have to perfectly represent the device's actual power usage. As long as it can roughly estimate the power consumption with the given intermediate values, a successful DPA attack can eventually be done with enough power traces. Kocher *et al.* showed a DPA attack on the DES in their original work, utilizing the difference in power consumption when a bit 1" is processed versus when a bit 0" is processed [1]. Due to the fact that the hypothetical power consumption value may be expressed in just two different states, this technique is referred to as a single-bit or binary power model [56] or binary power model [57]. Numerous alternative power models, including as the Hamming weight model [2], the Hamming distance model [3], and the zero-value model [57], can be used to more precisely characterize the power consumption of the targeted device. The Hamming weight model implies that the power consumed by the targeted device when processing a multiple-bit intermediate value is proportionate to the number of 1's included in that intermediate value. The Hamming distance model takes into account the power consumption of the targeted device as a result of the intermediate value transition between different states. It should be proportionate to the number of bits that have been flipped between two intermediate value states. The zero-value model presupposes that the input data zero consumes substantially less energy than all other values.

Finally, the attacker employs a statistical function to compare the potential power consumption values associated with each key hypothesis to the actual measured trace data at each location along the trace's length. In other words, each hypothetical power consumption matrix column is statistically compared to a power trace data matrix column. All statistical comparison results can be arranged in a matrix of dimensions $\boldsymbol{K} \times \boldsymbol{T}$, where indices of each element respectively indicate the associated hypothesis subkey value and position of the involved data point in the power trace.

3.2 Correlation Power Analysis Attack

Brier *et al.* originally presented the Correlation Power Analysis attack in 2004, which makes use of the Hamming Distance power model and Pearson's Correlation Coefficient equation [3] to improve conventional DPA attacks. CPA attacks, like other DPA attacks, employ the same general attack methodology. As is the case with DPA attacks, a single CPA attack reveals only a single-byte subkey. Hence, at least sixteen attacks are required to recover the whole 16-byte secret key utilized in the AES-128 algorithm. The practical attack steps of a CPA attack is described as follows.

- **Step 1:** Selecting an intermediate value of the cryptographic algorithm. For example, when attacking an AES implementation, there are several options of intermediate value. We can arbitrary select the first round's S-Box substitution outputs as the intermediate value.

- **Step 2:** Collecting the target device's power consumption data while it is encrypting random plaintext inputs. The accompanying power trace for each encryption should span the whole encrypting interval. The length of a power trace is indicated by the symbol $\boldsymbol{T}$. Additionally, the random plaintext input associated with each encryption must also be logged. $\boldsymbol{D}$ denotes the number of measured traces. As a result of this step, we will acquire $\boldsymbol{D}$ plaintexts and $\boldsymbol{D}$ associated power traces. All measured power traces can be represented as a power trace data matrix of dimension $\boldsymbol{D} \times \boldsymbol{T}$.

- **Step 3:** Calculating the matrix of hypothetical intermediate values. We use D_d

and K_i to compute the hypothetical intermediate value matrix $I_{d,i}$ using (3.1). D_d represents the n^{th} byte in the d^{th} plaintext input, whereas K_i represents the i^{th} key hypothesis, with $K_i = i - 1; i \in (1, 256)$. To put it another way, we predict the value of K_i. Since the size of K_i is 01 byte, K_i can take any value in the range 0 to 255. With each hypothetical value of K_i and a known value of D_d, we compute a hypothetical value of S-box output in the 1st round.

$$I_{d,i} = Sbox(D_d \oplus K_i) \tag{3.1}$$

This calculation generates a hypothetical intermediate value matrix of dimension $\boldsymbol{D} \times \boldsymbol{K}$, where $\boldsymbol{D}$ denotes the number of measured traces and $\boldsymbol{K}$ denotes the number of subkey hypotheses.

- **Step 4:** Calculating a matrix of hypothetical power consumption values. By using the Hamming Weight power consumption model, each value in the hypothetical intermediate value matrix is translated to a hypothetical power consumption value. We obtain the hypothetical power consumption matrix of dimension $\boldsymbol{D} \times \boldsymbol{K}$ after mapping. Then, using (3.2), the hypothetical power consumption value is calculated.

  $$H_{d,i} = HW(I_{d,i}) \tag{3.2}$$

  where:

  $HW()$ is the Hamming weight function, which counts the number of bits “1” in the binary input.

  $I_{d,i}$ denotes the hypothetical intermediate value matrix obtained in step 3.

  $H_{d,i}$ denotes the hypothetical power consumption value matrix.

- **Step 5:** Comparing the hypothetical power consumption values with the measured power traces. Pearson’s Correlation Coefficient is utilized as the statistical function for comparing the hypothetical power consumption values for each key hypothesis to the actual acquired trace data at each point along the trace length. All statistical comparison results may be represented in a matrix of size $\boldsymbol{K} \times \boldsymbol{T}$, where the indices of each element correspond to the associated hypothesis subkey value and location of the corresponding data point in the power trace. The indices of the element with the greatest value in the comparison result matrix indicate which hypothesis subkey is most likely to be the correct subkey and the time instant when the chosen intermediate value was processed. The Pearson's Correlation Coefficient can be calculated by using (3.3).

$$r_{i,j} = \frac{\sum_{d=1}^{D}(H_{d,i} - \overline{H_i}) \cdot (T_{d,j} - \overline{T_j})}{\sqrt{\sum_{d=1}^{D}(H_{d,i} - \overline{H_i})^2 \cdot \sum_{d=1}^{D}(T_{d,j} - \overline{T_j})^2}} \tag{3.3}$$

in which:

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

$\overline{T_j}$ is the mean value of all j^{th} sample points in D traces.

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

$\overline{H_i}$ is the mean value of the i^{th} column in the $H_{d,i}$ matrix.

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

D is the number of traces that will be used in analysis.

The correlation coefficients $r_{i,j}$ indicate the linear relationship between two variables, $H_{d,i}$ and $T_{d,j}$. The greater the absolute value of $r_{i,j}$, the more closely these two variables correlate. As a result, the correlation coefficients with the greatest absolute value will provide the most precise subkey prediction.

Moreover, Pearson correlation coefficients are scale independent of the two variables being compared. Thus, the power trace might be represented in the oscilloscope's raw data format rather than the scaled format of the voltage level in mV. This property may aid in the elimination of rounding errors in subsequent analysis.

However, when all D power traces are investigated, utilizing (3.3) requires that all D power traces be entered into the calculation simultaneously and delivers just the final correlation coefficient matrix $r_{i,j}$. If the D traces alone are insufficient to determine the secret key, additional traces must be collected and analyzed. Then, the entire analysis of all obtained traces must be repeated, including the D examined traces. Fortunately, equation (3.3) could be presented in an alternate form, which is shown in (3.4), where the correlation coefficient matrix is updated with every additional trace. Hence, equation (3.4) is more preferred in practice.

$$r_{i,j} = \frac{D\sum_{d=1}^{D} H_{d,i}T_{d,j} - \sum_{d=1}^{D} H_{d,i} \sum_{d=1}^{D} T_{d,j}}{\sqrt{((\sum_{d=1}^{D} H_{d,i})^2 - D\sum_{d=1}^{D} H_{d,i}^2)((\sum_{d=1}^{D} T_{d,j})^2 - D\sum_{d=1}^{D} T_{d,j}^2)}} \tag{3.4}$$

3.3 Deep Learning Based Side Channel Attack

Maghrebi *et al.* proposed the Deep Learning-based Side-Channel Analysis (DL-SCA) attack in 2016 as a state-of-the-art profiled power analysis attack [15]. They proved the DL-strength SCA's by demonstrating its ability to quickly crack both unprotected AES implementations and protected AES implementations with masking countermeasures. Since then, several related studies have been released that demonstrate more sophisticated characteristics of the DL-SCA attacks [16, 17, 20]. Cagli *et al.* effectively demonstrated in 2017 that DL-SCA attacks may be utilized against jitter-based countermeasures, which is a hiding countermeasure. In 2018, Benadjila *et al.* developed a profiled DL-SCA attack that makes use of a convolutional neural network (CNN) and is capable of exploiting severely desynchronized power traces [17]. Timon *et al.* demonstrated in 2019 that DL-SCA attacks may also be used in non-profiled scenarios [20].

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

- **Step 1:** The adversary must have complete access to a profiling device, which is an open, exact replica of the targeted device. The power traces of the profiling device are recorded while various inputs are encrypted/decrypted using different secret keys.

- **Step 2:** Each trace in the profiling traces is labeled with the input plaintext and secret key that relate to it.

- **Step 3:** The labelled traces are used as the training data for a Deep Learning process. As a result of this process, a trained deep learning network is created that is capable of classifying each power trace according to the key utilized.

- **Step 4:** The attackers acquire additional power traces from the targeted device. This collection is referred to as attacking traces. The attackers then utilize the trained network acquired in step 3 to determine the secret key value used by the targeted device based on the classification results of the network.

In [15], authors conducted four different DL-SCA attacks, including Auto Encoder (AE) based attack, Convolutionnal Neural Network (CNN) based attack, Long and Short Term Memory (LSTM) based attack, and Multilayer Perceptron (MLP) based attack. The architecture and the used parameters of deep learning networks used in these attacks are listed as follows.

- **Auto Encoder:**
  - Dense input layer: the number of neurons is the number of samples in the collected power trace
  - Dense hidden layer: 100 neurons
  - Dense hidden layer: 50 neurons
  - Dense hidden layer: 20 neurons
  - Dense output layer: 256 neurons
- **Convolutionnal Neural Network:**
  - Convolution layer: using 8 filters of length 16, Activation function is Rectified Linear Unit
  - Dropout
  - Max pooling layer with a pooling size of 2
  - Convolution layer: using 8 filters of length 8, Activation function is $tanh(x)$
  - Dropout
  - Dense output layer: 256 neurons
- **Long and Short Term Memory:**
  - LSTM layer: 26 units
  - LSTM layer: 26 units
  - Dense output layer: 256 neurons

- **Multilayer Perceptron:**
  - Dense input layer: the number of neurons is the number of samples in the collected power trace
  - Dense hidden layer: 20 neurons
  - Dense output layer: 256 neurons

The results from various experiments in [15] show that the Convolutionnal Neural Network outperformed other deep learning networks. Furthermore, in [17] Benadjila *et al.* proposed a Convolutionnal Neural Network architecture and demonstrated its ability to attack on severe misaligned power traces. Therefore, in this work, we adopt the best CNN architecture (CNN_{best}) proposed in [17] for further evaluation using profiled DL-SCA attacks since our proposed RDFS countermeasure also based on the severe misalignment of POIs in measured power traces. 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.

3.4 Test Vector Leakage Assessment

Key extraction attacks are a solid way to assess the security of a targeted cryptographic device. However, if the targeted device is secured against power analysis attacks, the number of traces necessary to effectively expose the secret key would grow considerably. That implies the volume of data to be evaluated is tremendous, and the analysis process would take an excessive amount of time to complete. In 2011, Goodwill *et al.* proposed a technique for evaluating side-channel leakage in a conformance style called Test Vector Leakage Assessment (TVLA) [58]. Later in 2013, Cooper *et al.* observed that performing TVLA for side-channel leakage detection is orders of magnitude quicker than completing realistic power analysis attacks [59]. TVLA's approach was to collect power consumption traces from the targeted device while performing cryptographic operations on a predefined set of input test vectors and then use statistical hypothesis testing to determine whether or not there is a sensitive intermediate value that has a significant effect on the measured traces. Goodwill *et al.* suggests several forms of leakage testing, each of which is focused on a potential leaking point in the cryptographic algorithm [58]. Cooper *et al.*, on the other hand, demonstrated that the non-specific test (also known as the fixed versus random test) is the most powerful. This non-specific test is capable of detecting a wide range of leakages with orders of magnitude fewer measurement traces than tests that focus on specific leakage spots. As a consequence, the non-specific test has been employed in a number of recent works to assess the security of side-channel attack-resistant devices [51, 43, 44, 45].

According to [59], details of the non-specific TVLA test procedure can be listed as follows.

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

Two sets of power traces (DataSet-1 and DataSet-2) must be gathered by monitoring the power consumption of the targeted device while encrypting it with AES-128 using a specified encryption key and a set of plaintext input.

For obtaining the DataSet-1, the encryption key is fixed to:

0x0123456789ABCDEF123456789ABCDEF0.

Then, $2n$ encryptions is performed on plaintext inputs $I_0, I_1, ..., I_{2n-1}$ 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.

The power trace acquisition for two data sets are randomly alternated to minimize any bias caused by measurement conditions such as temperature or electromagnetic interference noise. The length of each power trace can be denoted as L.

- **Step 2:** Applying Welch's t-test [60] on first $n/2$ traces from DataSet-1 and first $n/2$ traces from DataSet-2. The Welch's t-test equation is shown in (3.5).

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}} \tag{3.5}$$

$\overline{X_1}$ 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 $n/2$. Consequently, the output of this step is a t-score trace of length L. This t-score trace is denoted as T_{1st}.

- **Step 3:** Applying Welch's t-test on second $n/2$ traces from DataSet-1 and last $n/2$ traces from DataSet-2. The same Welch's t-test in Step 2 is applied to different sub-datasets. As a result, another t-score trace T_{2nd} of length L is obtained. It is important to repeat the Welch's t-test twice on distinct sub-datasets to reduce false positives in leakage detection. [58].

- **Step 4:** Comparing two t-score traces T_{1st} and T_{2nd}. The targeted device fails if both traces show a point that surpassed the ±4.5 range at the same time during the middle third of the AES operation.

It is critical to note that the Welch's t-test is used to examine the null hypothesis that the two sets of power traces (fixed-inputs and random-inputs) have equal means and variances. A large absolute value for the t-score implies a high degree of confidence in rejecting the null hypothesis. The limit of ±4.5 is selected such that the null hypothesis may be rejected with 99.99% confidence if the t-score exceeds it. If the computed t-score maintained in ±4.5 range, the null hypothesis is correct, and the targeted device is considered secure with power analysis attacks up to n power traces. In other words, the tester is incapable of detecting any sensitive intermediate value that has an effect on the power traces being measured.

Chapter 4

Exploiting Back-gate Bias As Countermeasure

4.1 Proposed Idea

By analyzing the power analysis attack premise, it is obvious that the attackers' success is reliant on the instantaneous power consumption being dependent on the intermediate values being processed. This data dependence is primarily induced by the switching operations of internal logic cells utilized to compute intermediate values. However, in very large-scale integrated circuits, these switching activities account for a negligible part of the dynamic power consumed by the device. The remains of the total power consumption is dynamic power generated by other logic cells' switching activities combined with static power consumption, which is primarily driven by leakage current. Fortunately, the FD-SOI technology in general, and the SOTB technology in particular, enable dynamic fine-tuning of both the leakage and drive currents of internal logic cells using the back-gate biasing approach. Therefore, back-gate biasing could be used to control the noise ratio of the power trace. Hence, we propose to use back-gate biasing in countermeasures against power analysis attacks.

The first proposed idea is to supply the back-gate bias of the targeted cryptographic device with a forward bias voltage. Applying forward back-gate bias voltage greatly increases the drive current and device's maximum operational clock frequency; it also significantly elevates the leakage current. However, because this leakage current is treated as noise in power analysis attacks, hence introducing forward back-gate bias increases the noise level and enhances device's resistance to power analysis attacks.

Additionally, statistical functions used in power analysis attacks are performed on a large set of power consumption traces in DPA attacks. If the noise level in each trace is a uniformly distributed random variable, the correlation between the measured power traces and the processing data is weakened even further.

However, the secret information may be discovered only by analyzing the power consumption at the exact moment that the intermediate values are being processed. Only the injection of random noise to that specific moment helps to resist power analysis attacks. Therefore, the second proposed idea is to randomly vary the back-gate bias of the targeted device in such a way that the noise level at that exact point in the set of power consumption trace is distributed equally and randomly. This proposed countermeasure is denoted as random dynamic back-gate biasing (RDBB). We chose to alter the back-gate bias after each encryption or decryption in our later experiments.

Both proposed countermeasures are applicable to any device equipped with back-gate bias controllability, as they rely on the back-gate biasing technique. In other words, regardless of the cryptographic algorithm or system architecture employed, all devices fabricated using FD-SOI or a related technology can benefit from the proposed countermeasures.

4.2 Related Works

The proposed back-gate bias countermeasures can be classified as hiding countermeasures. Both proposed techniques aim to randomly alter operating conditions of cryptographic devices to cause distortions in power traces. Numerous articles on the subject should be considered. For example, Yang *et al.* presented the random dynamic voltage frequency scaling (RDVFS) countermeasure in 2005. This countermeasure requires changing the clock frequency and supply voltage of the targeted device at random times while it is operating. In 2007, Baddam *et al.* demonstrated that using random dynamic voltage scaling (RDVS) alone might result in improved DPA resistance [47], because a change in frequency could be easily recognized by observing a power consumption trace. These countermeasures take advantage of the fact that dynamic power usage is related to both the clock frequency and the supply voltage's square. Furthermore, these authors claimed that the RDVFS countermeasure and RDVS countermeasure may be directly implemented to a customized ASIC or general microcontroller without altering the cryptographic algorithm or hardware design architecture. However, in their initial work, they employed solely software-simulated power consumption traces of cryptographic circuits to illustrate the success of their proposed countermeasures in experimental DPA attacks. While the simulation method is fast and flexible, it may neglect the serious issue of dynamically changing the supply voltage and clock frequency of a running device, which has a detrimental effect on the system's stability, particularly in complicated designs such as microcontrollers or CPUs. As a result, computation errors, read/write errors in registers, and even system failures will occur. Geng *et al.* published a more realistic evaluation of these countermeasures [61], in which the power traces used in their power analysis attacks are measured from an FPGA implementation of the S-Box substitution function. However, Geng *et al.* only mimicked the effects of RDVS by using post-measure processing with MATLAB. To avoid these limitations and deliver more trustworthy assessment results for our new proposals, a 32-bit RISC-V microcontroller fabricated in SOTB 65nm technology is used as the target device in later experiments. The power consumption traces will be measured when actual different random supply voltage and back-gate bias are applied to our microcontroller. This is the only method to account for the practical limitations of evaluation in existing related papers and achieve objective evaluation results.

4.3 Experiments

4.3.1 Experimental setup

4.3.1.1 Test device

Silicon-on-thin-BOX (SOTB) technology is a subset of the Fully Depleted Silicon on Insulator (FDSOI) family of technologies. It was initially developed to address the problems of standby power and characteristic variation in fabricated IC [40]. Figure 4.1 illustrates the cross-sectional view of a SOTB complementary metal oxide semiconductor (CMOS) device that contains triple wells, namely deep N-well, N/P-well, and N+/P+, shallow trench isolations (STIs), and ultra-thin buried oxide (BOX) layers. The SOTB technique differs from standard FD-SOI technology in that it utilizes an ultra-thin layer of buried oxide (BOX). The advantage of utilizing ultra-thin BOX is that it provides strong resistance to the short channel effect and suppresses the threshold V_{th} variation. Additionally, it permits the use of a broader range of back-gate bias voltages, resulting in far more efficient transistor's performance control [62]. As a result, SOTB devices can be employed in a wide variety of embedded applications, ranging from low-power IoT

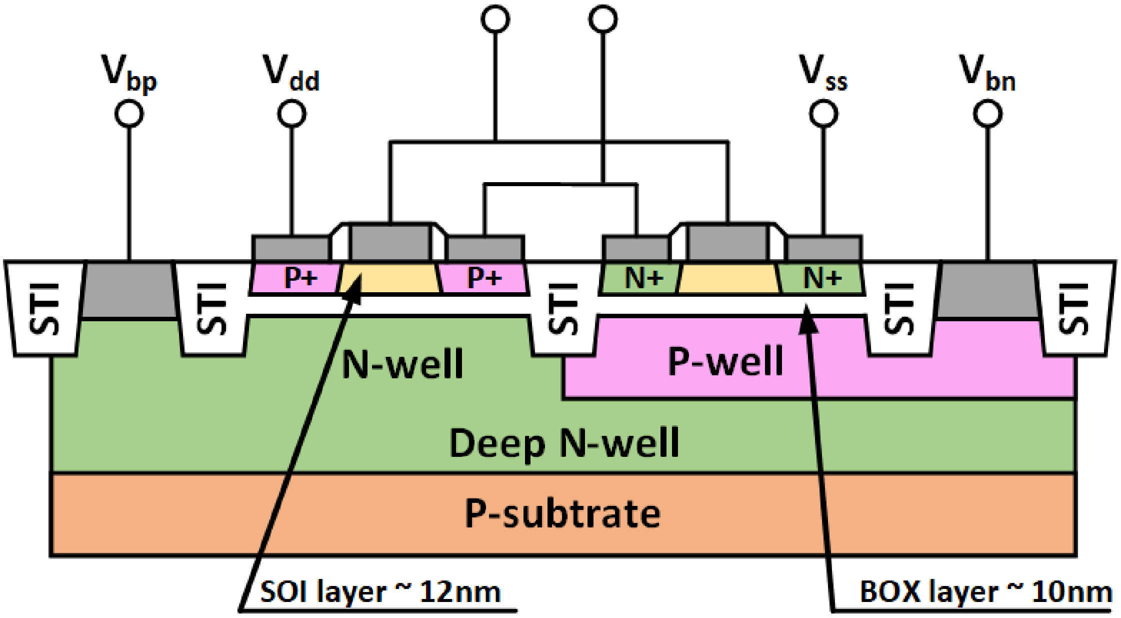


Figure 4.1: Cross-sectional view of the SOTB CMOS.

edge devices to high-performance circuits that utilize the forward back-gate bias voltage. Furthermore, SOTB devices are well-suited for low-voltage operation due to the low impurity concentration in the channel regions, which results in low variation [63]. Therefore, SOTB devices are the optimal choice for experimenting and evaluating the proposed countermeasures due to their ability to operate over a wide range of back-gate bias and supply voltage. In the subsequent DPA attack experiments, A 32-bit RISC-V microcontroller, which is designed and fabricated using the 65 nm SOTB process, is utilized as the target device. The back-gate bias of the target microcontroller can be supplied to apply the proposed countermeasures. DPA attacks are conducted when the target microcontroller functions with and without applying the proposed countermeasures. DPA attacks' results are used as evaluation metrics for the effectiveness of the proposed countermeasures.

The architecture of the 32-bit RISCV microcontroller under consideration is depicted in Figure 4.2. The microcontroller is consisted of a core CPU, 8 KB of boot ROM, 64 KB of on-chip SRAM memory, a UART controller, an SPI controller, a timer, and GPIOs. The boot ROM includes 1 KB of stack memory in SRAM and 7 KB of hard-code in combinational logics. The core CPU is the VexRiscv CPU with RV32IM ISA extensions [64]. RV32IM ISA extensions mean that this microcontroller supports 32-bit RISC-V Base Integer Instruction Set and 32-bit RISC-V Standard Extension for Integer Multiplication and Division. The SPI controller is included to facilitate the use of the

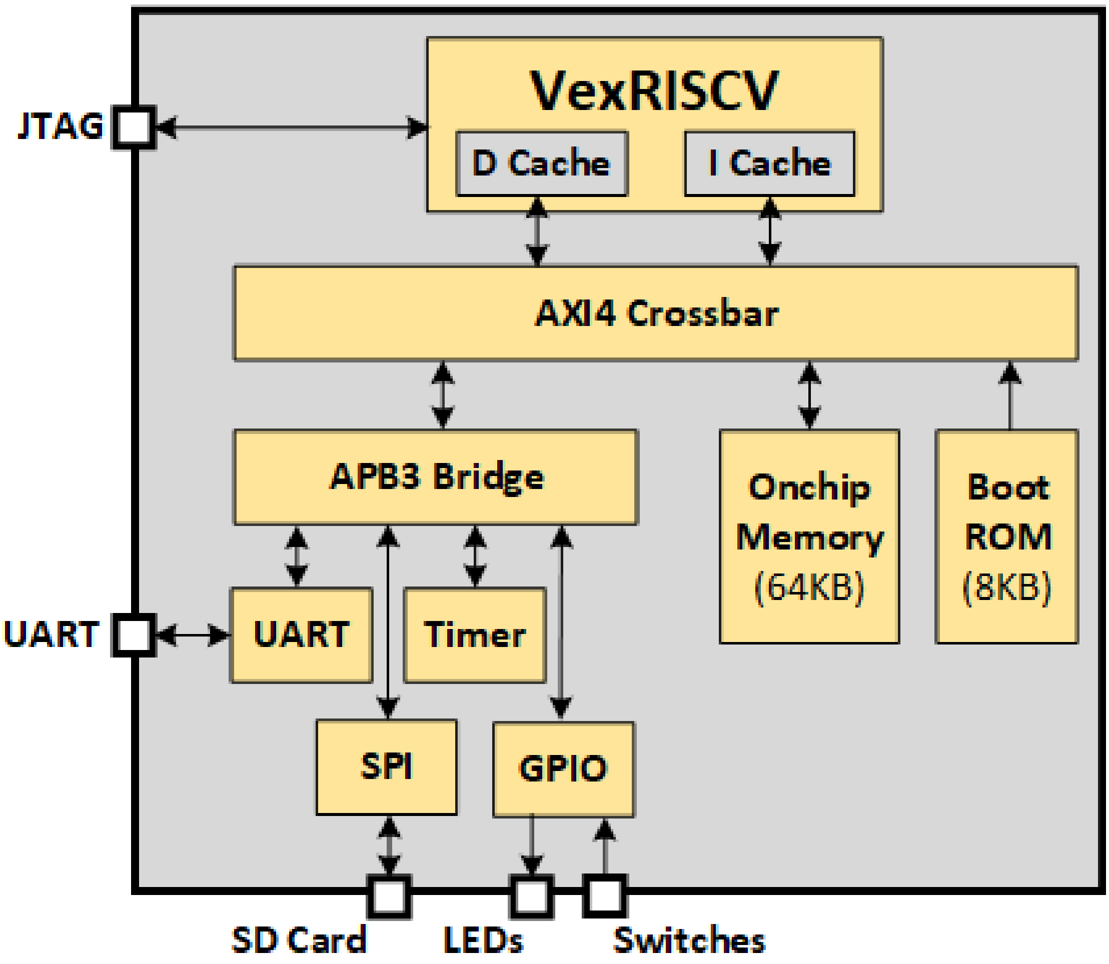


Figure 4.2: The microcontroller architecture.

external SD card. Any algorithm can be compiled as firmware, stored in external SD card and executed by the microcontroller following the microcontroller's boot procedure. The GPIO contains sixteen output pins and sixteen input pins. The input-GPIOs are drived by sixteen in-off switches. The output-GPIO drives sixteen LEDs. The UART controller is added to the target 32-bit RISCV microcontroller to communicate with other UART devices. The microcontroller can also be debugged by using its JTAG port.

The target microcontroller's back-gate bias can be adjusted to favor low-power or high-performance operations [41]. The microcontroller's back-gate bias can be adjusted between -2.0V and 2.0V. The internal logic core's supply voltage also can range between 0.5 and 1.2 V, meanwhile the I/O banks' supply voltage is fixed at 3.3V. In general, this F_{max} value grows linearly as V_{DD} increases. F_{max} is also increased if the back-gate bias V_{BB} is increased in a range of -2.0V to 0.8V. When $V_{BB} \geq$ 1.6V is used, hardly no improvement in F_{max} occurs. The microcontroller's highest clock frequency is F_{max} = 156 MHz when the core supply voltage V_{DD} is 1.2V and the back-gate bias V_{BB} is 1.6V. The microcontroller's active power density likewise varies in response to variations in V_{DD} and V_{BB}. In general, it also increases linearly as V_{DD} increases. Additionally, forward back-gate bias considerably increases the power density. The greatest measured power density is 289 μW/MHz when V_{DD} = 1.2V and V_{BB} = 2.0V. However, the power density experiences only a slight reduction when the microcontroller is biased at reversed back-gate bias. The lowest power density of 33.4 μW/MHz was achieved at the lowest operating point of 0.5V V_{DD} with 0.8V V_{BB}.

Figure 4.3 illustrates the PCB platform for using the target microcontroller. The PCB board is integrated with a chip socket for hosting the target microcontroller, an USB-to-UART interface, a clock generator chip for generating a programmable clocks, and additional peripherals such as an SD-card socket, a JTAG header, sixteen LEDs, and sixteen switches. By using or not using the power jumpers, the power supply, the back-gate bias can be pulled directly from the USB interface or from external power sources. Additionally, the working clock can also be supplied externally via the SMA connector.

4.3.1.2 Automatic power trace measuring system

Power analysis attacks demand the acquisition of a large number of power traces. The more power traces acquired by the attacker, the more trustworthy and accurate the attack results will be. Thus, an automated power trace measuring system is developed. The system's is depicted in Figure 4.27. This figure shows all of the required equipment of the our automated power trace measuring system, including as follows:

1. *Targeted device:*

   The target device is a 32-bit RISC-V microcontroller built using 65-nm SOTB technology. The target microcontroller's clock frequency is fixed at 12 MHz. An open-source C implementation of the AES-128 algorithm is compiled and implemented into the firmware of the target microcontroller [65, 66]. The firmware also enables our target device to interface with the control PC via a UART port. The target device receives data, encrypts or decrypts it, and then sends back the encryption/decryption results. The status signal is generated using a GPIO pin on the targeted device. It indicates whether or not the AES-128 algorithm is in use.

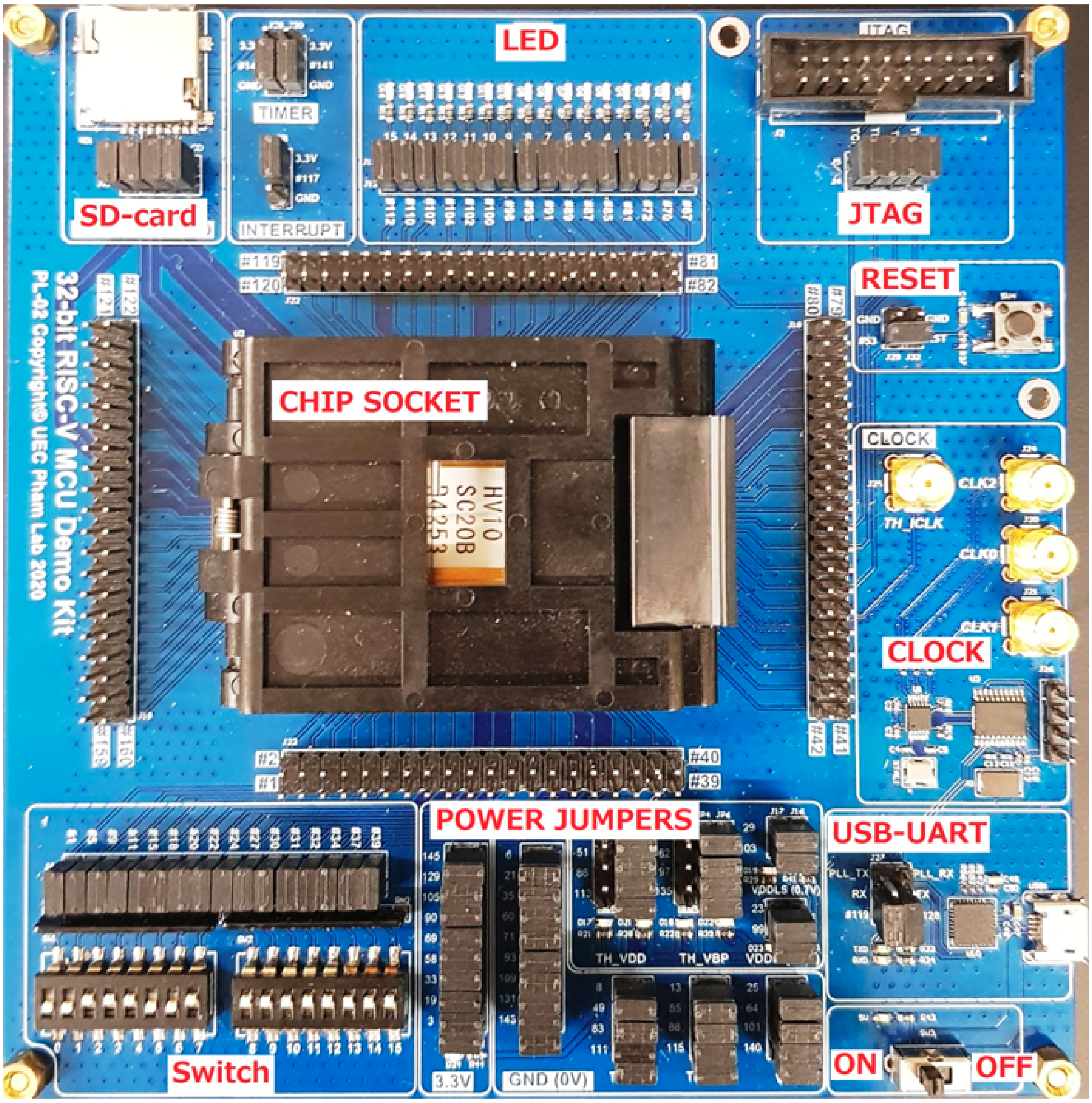


Figure 4.3: PCB platform for testing the target microcontroller [41].

2. *Power Supply:*

   The ADCMT-6541 power supply unit is utilized to create and supply DC voltages for the core V_{DD} and bias voltage V_{BB} of the targeted microcontroller. It includes a USB interface for remote programming. Through this USB interface, the ADCMT-6541 is connected to and controlled by the control PC.

3. *Oscilloscope:*

   The Tektronix MSO2024B oscilloscope is used to capture power traces generated by the targeted device as it performs cryptographic operations. Four analog channels are included in this oscilloscope. The highest sampling rate is 1 GS/s and

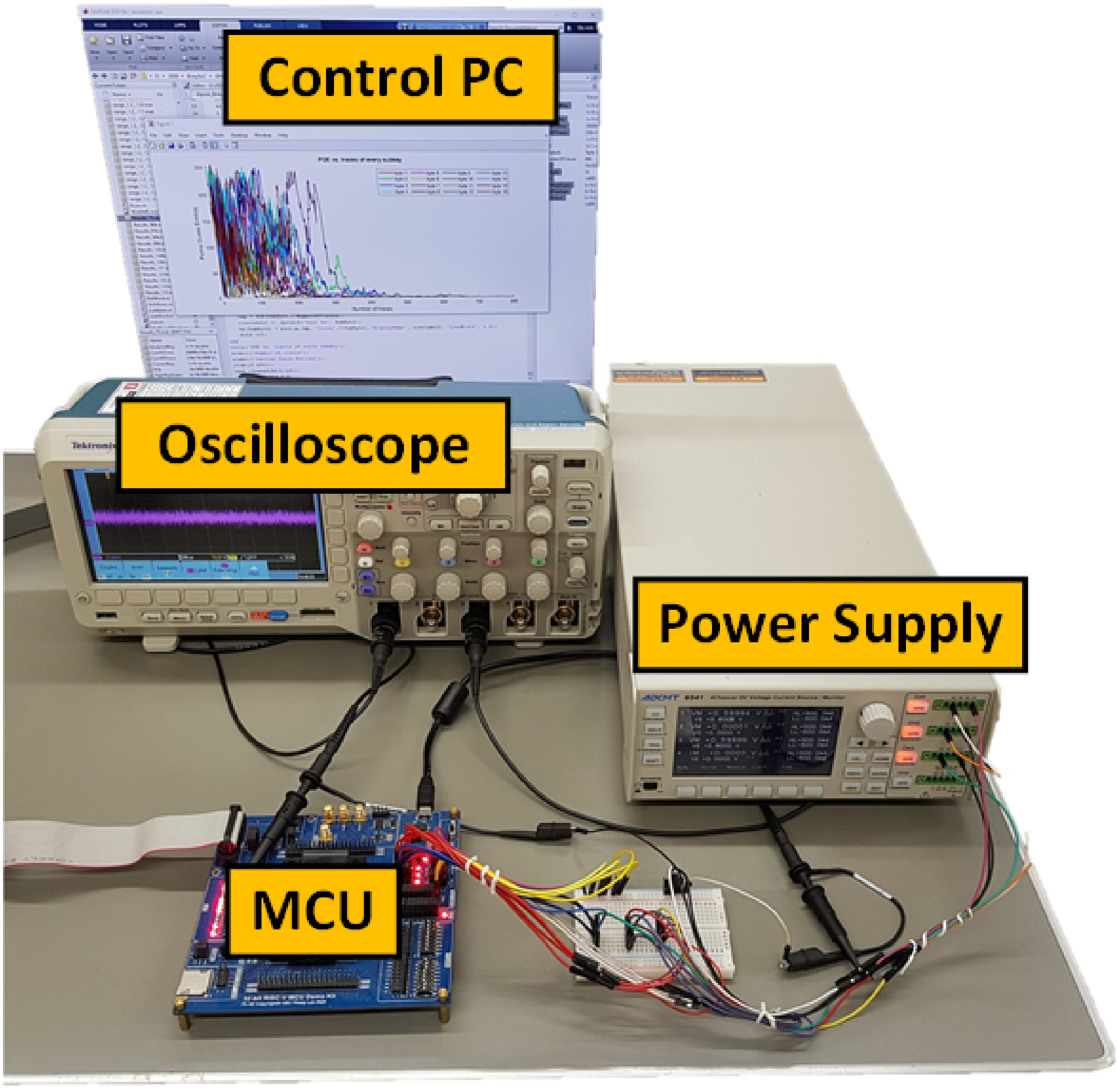


Figure 4.4: The developed automated trace acquiring system.

the maximum bandwidth is 200 MHz. The measurement is made using two Tektronix passive probes TPP0200. One probe is connected to the target device's status signal and serves as the oscilloscope acquisition trigger signal. The other probe acquires the analog signal from the targeted device's core V_{DD} node. Additionally, this oscilloscope may be remotely controlled via VISA (Virtual Instrument Software Architecture) interface from the control PC.

4. *Control PC:*

   A PC is executed the control script and control the entire auto-measuring system. It transmits configuration data to the oscilloscope and power supply; obtains and checks cryptographic output data from the targeted device; and acquires and saves

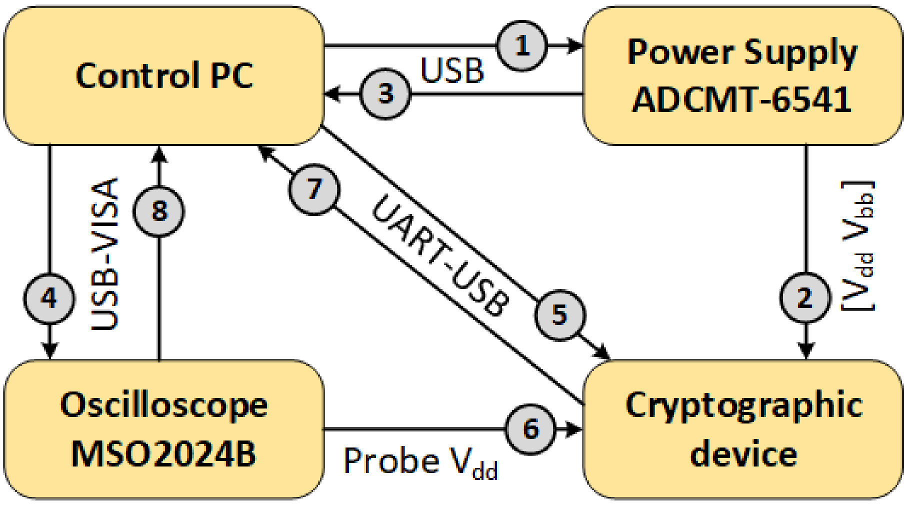


Figure 4.5: Operation steps of the automated trace acquiring system.

measured data from the oscilloscope and power supply.

At first, the control PC performs the initialization, which includes connecting to the oscilloscope and the power supply via USB ports. The initialization process also includes connecting to the target microcontroller via UART-USB interface. The control PC configures the initial parameters of the oscilloscope (trigger type, sampling rate, bandwidth, measuring range, and so on) and the power supply (initial DC output voltages). Following that, the auto-measuring system loops through multiple steps to obtain the appropriate amount of power traces. These steps are labeled (1) through (8) in Figure 4.5:

1. The targeted microcontroller finished its boot procedure and ready to execute the AES-128 algorithm. The control PC determines the values of core V_{DD} and bias voltage V_{BB} and commands the ADCMT-6541 power supply to generate corresponding output voltages.

2. The power supply issues the received commands and then generates and applies desired V_{DD}, V_{BB} to the targeted microcontroller.

3. The power supply measures the actual voltages applied to the targeted microcontroller and sends the measured values back to the control PC for verification.

4. The oscilloscope receives the invoke acquisition command from the control PC. The oscilloscope enters the single acquisition mode and waits for the target microcontroller to generate a trigger signal.

5. The control PC extracts a 16-byte plaintext from the DPA Contest v2 public database's randomly generated plaintext list and delivers it to the targeted device [67]. Additionally, the microcontroller receives the 16-byte plaintext and echoes back to the control PC in order to verify that the plaintext data transmission was successful. Following that, both the control PC and the targeted device encrypt their copy of plaintext using AES-128 algorithm with Electronic code-book (ECB) encryption mode. The encryption key is a predetermined 16-byte key. Moreover, when encryption begins, the microcontroller changes the GPIO pin from logic "0" to logic "1" and switch it back to logic "0" when encryption is completed.

6. If the oscilloscope successfully detect the trigger signal from the GPIO pin, it start to acquire the voltage fluctuation of the microcontroller's core V_{DD} while the encryption is being executed.

7. When the encryption is complete, the microcontroller transmits the 16-byte ciphertext output back to the control PC. The control PC compares the received ciphertext to its encryption results to guarantee that there is no error in the microcontroller's encryption procedure.

8. The measured waveform is sent to the control PC by the oscilloscope. If all of the checking conditions in steps 3, 5, and 7 are met, the control PC saves the measured waveform as a power trace, as well as the corresponding plaintext and ciphertext, for later analysis.

A passive probe may only record the voltage-level fluctuation of the V_{DD} node but not the instantaneous power consumption of the targeted device. A tiny resistor must be inserted in the core V_{DD} line of the targeted device to properly assess the instantaneous power usage. A differential probe should be used to measure the voltage difference across this resistor, which is proportional to the current consumed by the targeted device. The acquired voltage drop and resistor resistance can be used to calculate the actual instantaneous power consumption. Many earlier publications have proven that power traces measured with a passive probe still contain nearly the same information that the attacker requires for statistical analysis, [57, 56]. Furthermore, in practice, the differential probe is a high-end, expensive equipment that may not be available. Besides , the goal of these tests is to gain a fair comparison of various scenarios in which countermeasures against power analysis attacks are deployed to the targeted device. Thus, as long as the same experimental conditions and equipment are employed for all of subsequent power analysis attacks, adopting the passive probe approach is an acceptable choice in our automated power trace measuring system.

In our target microcontroller, there is no on-chip voltage regulator. The internal logic core's supply voltage and back-gate bias are applied directly via associated pins. Thus, by utilizing the automatic power trace measurement system, the supply voltage and back-gate bias of the target microcontroller can be modified dynamically. Step 1 of the iteration of the automatic power trace measurement system is modified. After acquiring a power consumption trace of the target microcontroller, the control PC creates fresh values for the back-gate bias V_{BB} and supply voltage V_{DD}. These values could be fixed numbers or uniformly distributed pseudorandom values created by the MATLAB function *randi()* . New V_{DD} or V_{BB} values are supplied to the microcontroller, and power traces of the new setup can be obtained as usual. The conditional tests in steps 3, 5, and 7 verify that the target microcontroller is stable and functions appropriately even when the supply voltage and back-gate bias are modified dynamically.

4.3.1.3 Power trace analysis

A MATLAB application is developed for statistically analyzing power traces. The MATLAB application uses measured power traces and either plaintexts or ciphertexts as inputs and then outputs an analyzed key. Both plaintexts and ciphertexts related to the measured power traces are available. As a result, we have several candidates for the intermediate value. The first round's S-Box substitution output is chosen as the intermediate value for the DPA attack. The same set of inputs is used to investigate all 16 subkeys at the same time. The first step in the DPA attack on the n^{th} byte of the secret key is to compute the hypothetical intermediate value matrix. Each row of the hypothetical intermediate value matrix comprises all potential S-Box output values in the first round, which correspond to all possible values of the n^{th} subkey. This matrix can be computed using (4.1).

$$I_{d,i} = Sbox(D_d \oplus K_i) \tag{4.1}$$

where:

D_d denotes the n^{th} byte of the d^{th} plaintext input.

K_i denotes the i^{th} key hypothesis, with $K_i = i - 1; i \in (1, 256)$.

$Sbox()$ denotes the SubBytes operation in the AES-128 algorithm.

$I_{d,i}$ denotes the hypothetical intermediate value based on the d^{th} plaintext and i^{th} key hypothesis.

The second step is to derive the hypothetical power consumption value. The device under consideration is a general-purpose microcontroller that runs a software implementation of the AES-128 algorithm. Because a microcontroller's instantaneous power consumption should be inversely proportional to the Hamming weight of the executed data, the Hamming weight of the intermediate value is used as the power model [57]. The hypothetical power consumption value is calculated as follows:

$$H_{d,i} = HW(I_{d,i}) \tag{4.2}$$

in which:

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

$I_{d,i}$ denotes the hypothetical intermediate value based on the i^{th} key hypothesis and the d^{th} plaintext.

$HW()$ denotes the Hamming weight function, which counts the number of bits "1" in the input binary number.

The analytical application then performs a statistical comparison between the hypothetical power consumption values associated with each key hypothesis and all collected power traces at every single sampling point as the final step in a DPA attack. The Pearson correlation coefficient is used as a comparison metric, and it can be calculated with (4.3).

$$r_{i,j} = \frac{\sum_{d=1}^{D}(H_{d,i} - \overline{H_i}) \cdot (T_{d,j} - \overline{T_j})}{\sqrt{\sum_{d=1}^{D}(H_{d,i} - \overline{H_i})^2 \cdot \sum_{d=1}^{D}(T_{d,j} - \overline{T_j})^2}} \tag{4.3}$$

in which:

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

$\overline{T_j}$ is the mean value of all j^{th} sample points in D traces.

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

$\overline{H_i}$ is the mean value of the i^{th} column in the $H_{d,i}$ matrix.

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

D is the number of traces that will be used in analysis.

The correlation coefficients $r_{i,j}$ represent the linear correlation between the two variables under consideration, i.e., $H_{d,i}$ and $T_{d,j}$. The greater the absolute value of $r_{i,j}$, the better the match between these two variables. As a result, the correlation coefficients with the greatest absolute value will provide the best subkey guess. Furthermore, the Pearson correlation coefficients are unaffected by the scale of the two variables under consideration. Therefore , the power trace could be expressed in the oscilloscope's raw data format rather than the scaled format of the voltage level in *mV*. This property may aid in the elimination of rounding errors in later computation.

When all D power traces are evaluated, (4.3) requires all D power traces to be input into the calculation at once and returns only the final correlation coefficient matrix $r_{i,j}$. If D traces itself are insufficient to uncover the correct subkey, more traces must be measured and analyzed. T he whole analysis of all acquired traces, including the D examined traces, must then be redone. Equation (4.3) can be given in an alternate form, as illustrated in (4.4) , in which the correlation coefficient matrix is updated with each additional trace. This alternative form of Pearson's correlation coefficient is used in our MATLAB program.

$$r_{i,j} = \frac{D\sum_{d=1}^{D} H_{d,i}T_{d,j} - \sum_{d=1}^{D} H_{d,i}\sum_{d=1}^{D} T_{d,j}}{\sqrt{((\sum_{d=1}^{D} H_{d,i})^2 - D\sum_{d=1}^{D} H_{d,i}^2)((\sum_{d=1}^{D} T_{d,j})^2 - D\sum_{d=1}^{D} T_{d,j}^2)}} \tag{4.4}$$

Lastly, several criteria for evaluating the effectiveness of the DPA attack can be calculated. These evaluation criteria are as follows:

1. Partial guessing entropy (PGE) [68]: When the statistical comparison data of all key hypotheses are sorted from highest to lowest, PGE is the top-down ranking of the correct subkey. When the PGE hits 0, the actual subkey has the largest statistical comparison result among all of subkey hypotheses, and the attack succeeds in recovering the subkey from the input set of power traces.

2. The correlation coefficient value of the correct subkey hypothesis.

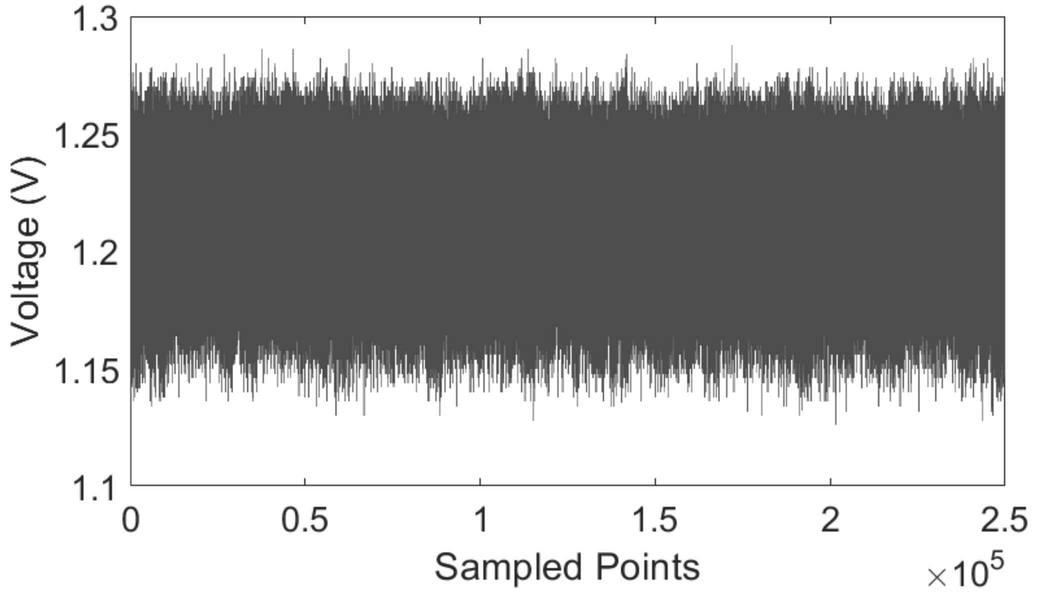


Figure 4.6: Example of a power trace at 0V V_{BB} and 1.2V V_{DD}.

3. The minimal number of power consumption traces required to reveal the secret subkey successfully.

4.3.2 Experimental results

4.3.2.1 Fixed supply and back-gate bias voltage

We attempt multiple DPA attacks on the targeted microcontroller under various conditions using the experimental setup provided in Section 3.3.1. To begin, the microcontroller is set to function with V_{DD} = 1.2V and no back-gate bias, i.e., V_{BB} = 0V. In this scenario, the results of the DPA attacks are utilized as a reference in later comparisons.

Figure 4.6 is an example of a raw power trace measured when the supply voltage V_{DD} = 1.2V and no back-gate bias is applied while the targeted microcontroller is performing AES-128 encryption. The power trace demonstrates that the voltage level of the V_{DD} node swings between 1.2V ± 50 mV. When ten rounds of AES-128 encryption are executed on a typical microcontroller, a series of nine similar patterns followed by a slightly different pattern is predicted to be noticed in the power trace. However, there is no such pattern in Figure 4.6, indicating that there is a substantial amount of noise in the power trace in this practical experiment. As a result, carrying out practical DPA attacks against the targeted 32-bit RISC-V microcontroller will be extremely difficult.

All 16 bytes of the encryption key are successfully exposed using the MATLAB analysis tool described in Section 3.3.1.3. The correlation coefficients versus a plot of the number of examined traces of all subkey hypotheses, which correspond to all sixteen subkeys, are shown in Figure 4.7. The graphs for the sixteen correct subkey hypotheses are drawn in sixteen different colors, whereas the graphs for the other incorrect subkey hypotheses are plotted in light gray. The correlation coefficient of incorrect hypotheses (light gray lines) always drops when more power traces are examined, as demonstrated in this figure. In contrast, after analyzing enough power traces for each subkey, only the correlation coefficient of the correct hypothesis will settle out at a larger value than the correlation coefficients of the other false hypotheses. The correlation coefficients for the correct hypotheses range from 0.193 to 0.285, with an average of 0.245. On the same set of measured power traces, each subkey is attacked using the same analytical procedure. However, the results of this experiment demonstrate that due to the substantial amount of noise in the power traces, several subkeys have much lower correlation values for the correct hypothesis. Therefore, they were more difficult to uncover.

Figure 4.8 depicts the plot of the partial guess entropy versus the number of traces for all 16 subkeys. It demonstrates that after examining around 755 traces, the PGE of all 16 subkeys is equal to zero. That is, in this case with no back-gate bias and a 1.2V supply voltage, the attacker needs at least 755 power traces to successfully find the actual encryption key. Furthermore, because the correlation coefficient of the correct hypothesis of each subkey varies significantly, the least number of traces required to expose each subkey varies as well. The DPA attack results for this test scenario indicate that analyzing a subkey requires an average of 389 power traces to be successful.

Similarly, other back-gate bias and supply voltage configurations are applied to the targeted microcontroller, and DPA attacks are conducted for each circumstance. The supply voltage V_{DD} is either 0.6 V, 0.9 V, or 1.2V in each case, whereas the back-gate bias V_{BB} ranges from -2.0V to 2.0V with a 0.4V step. Several comparison criteria are used to compare the outcomes of these DPA attacks:

1. minimal number of traces required to recover the full 16-byte key. or also known as the minimal MTD.
2. average of the 16 minimal required number of traces corresponding to 16 subkeys , or also known as the average Measurement to Disclosure (MTD).

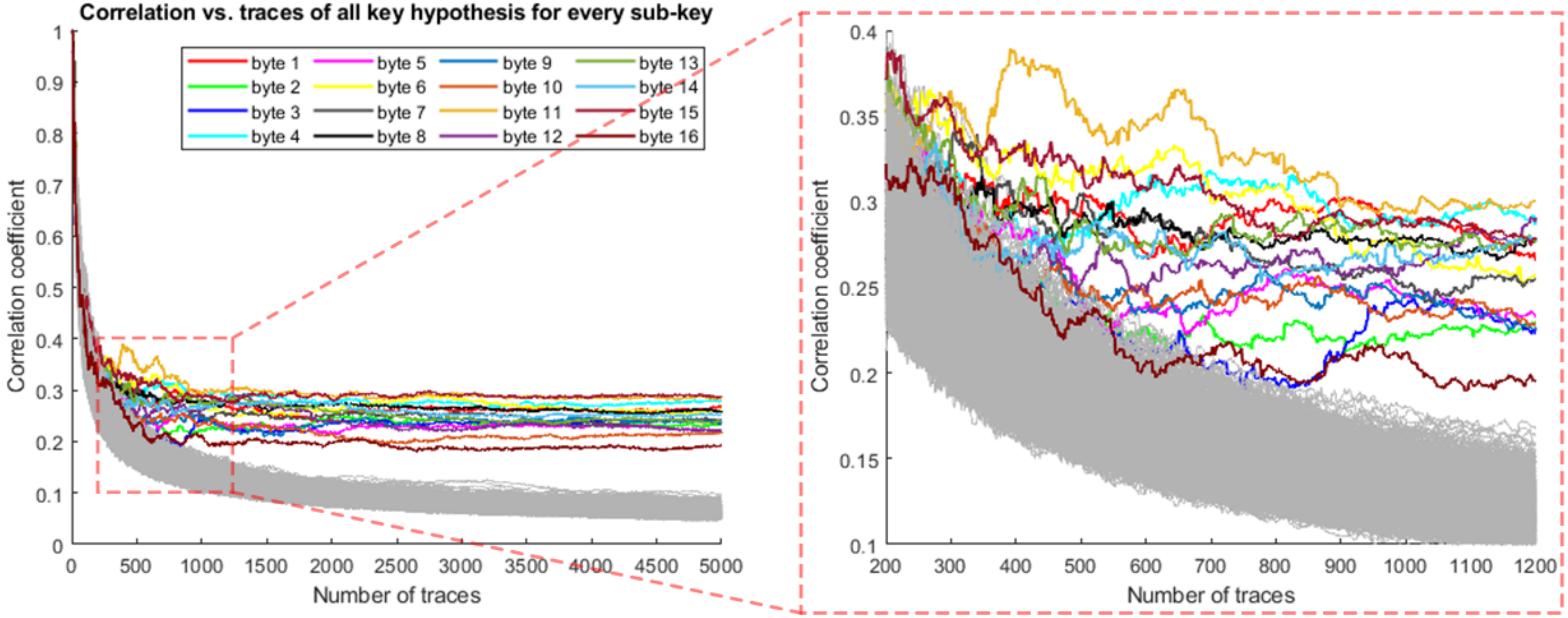


Figure 4.7: DPA Attack results of all subkey (Correlation versus analyzed traces) at 0V V_{BB} and 1.2V V_{DD}.

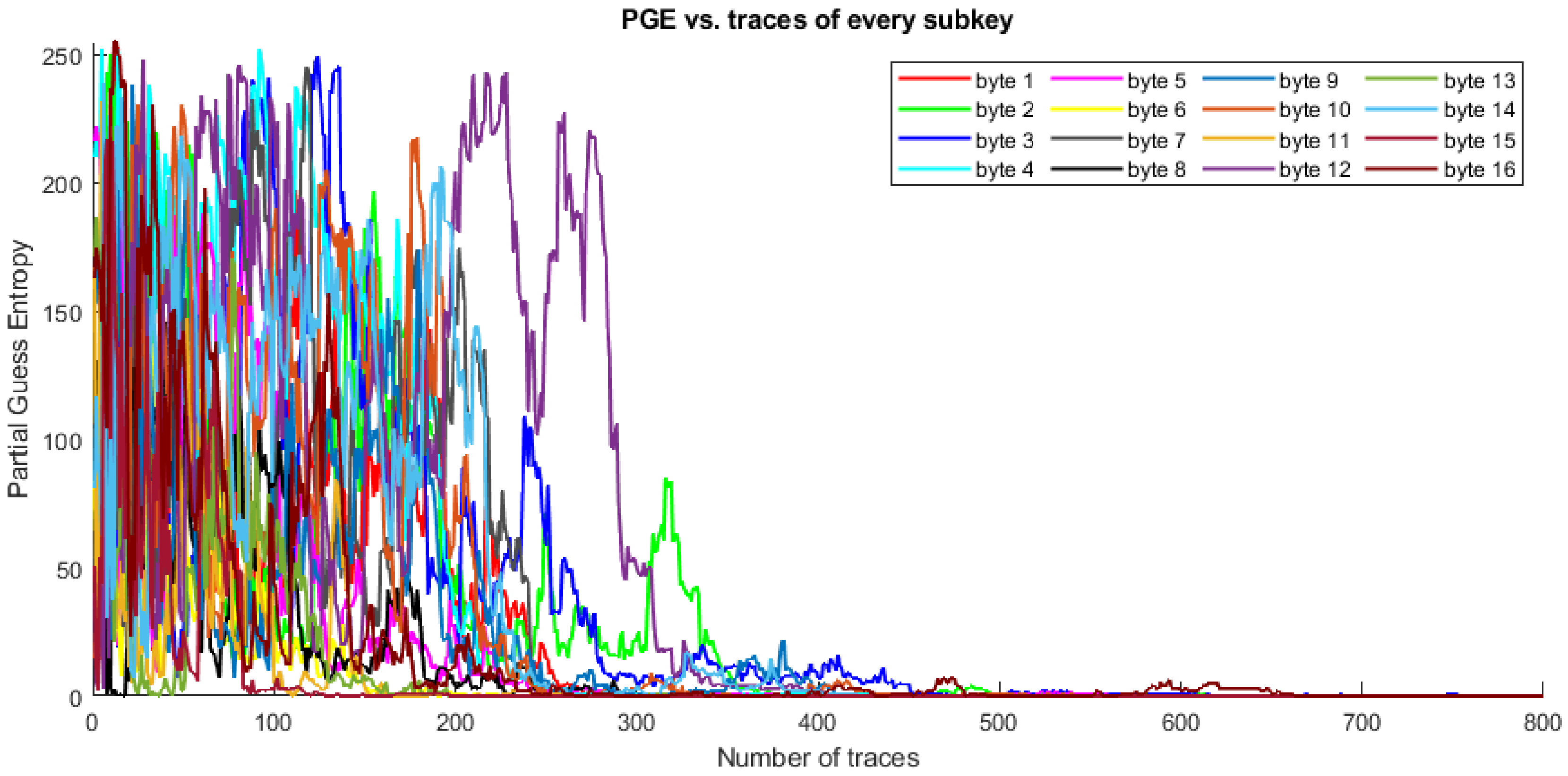


Figure 4.8: DPA Attack results of all subkey (PGE versus analyzed traces) at 0V V_{BB} and 1.2V V_{DD}.

3. average of the correlation coefficients of 16 correct hypotheses corresponding to 16 subkeys.

These DPA attack results are shown in Figure 4.9, Figure 4.10 and Figure 4.11, in that order. However, as mentioned by [41], the possible bias voltage range is limited when a lower supply voltage is employed. As a result, there are a total of 27 possible configurations. When reverse bias voltage is used, the correlation coefficients tend to remain at the same magnitude as when no bias voltage is used, as illustrated in Figure 4.9. Furthermore, as the targeted microcontroller's bias voltage goes deeper into the forward bias voltage region, the correlation coefficients reduce. Moreover, with the fixed bias voltage, lowering the supply voltage also reduces the correlation coefficients. The reason behind it is that the noise tolerance is poor at lower supply voltages. Particularly in our practical experiments, the impacts of interference and electronic noise in measuring power traces are more intense. Additionally, the order of reduction in the correlation coefficient caused by forward bias voltage is slightly lower than at a lower supply voltage. At a supply voltage of 1.2V, the correlation coefficient reduces by approximately 0.13 in magnitude, from a peak of nearly 0.25 to a low of more than 0.12. Meanwhile, at 0.9V and 0.6V supply voltages, the coefficient falls in magnitude by 0.11 (from 0.2 to 0.09) and 0.09 (from 0.15 to 0.06), respectively. The graphs for the required number of traces in Figure 4.10 and Figure 4.11 show similar trends since the required number of traces is inversely proportional to the correlation coefficient. As a result, introducing a deep forward bias voltage and lowering the supply voltage will improve the DPA resistance of the chosen microcontroller. The best DPA resistance is found at the deep forward bias voltage V_{DD} of 2.0V and the lowest supply voltage V_{DD} of 0.6V. When compared to the previous reference example, the average correlation coefficients in this best scenario reduce from 0.245 to 0.065, and the number of traces necessary to reveal the 16-byte encryption key increases from 755 to 10,925, an approximately 14.5-fold increase.

Despite the fact that forward and reverse bias voltage have different impacts on device performance and power consumption characteristics, the experimental results reveal that only forward bias voltage affects the DPA resistance of the targeted device. The reason is that in DPA attacks, the fraction of dynamic power linked to intermediate value processing relative to total power consumption, which is an exploitable leakage source for attackers, must be considered. It is quite difficult to precisely measure dynamic power

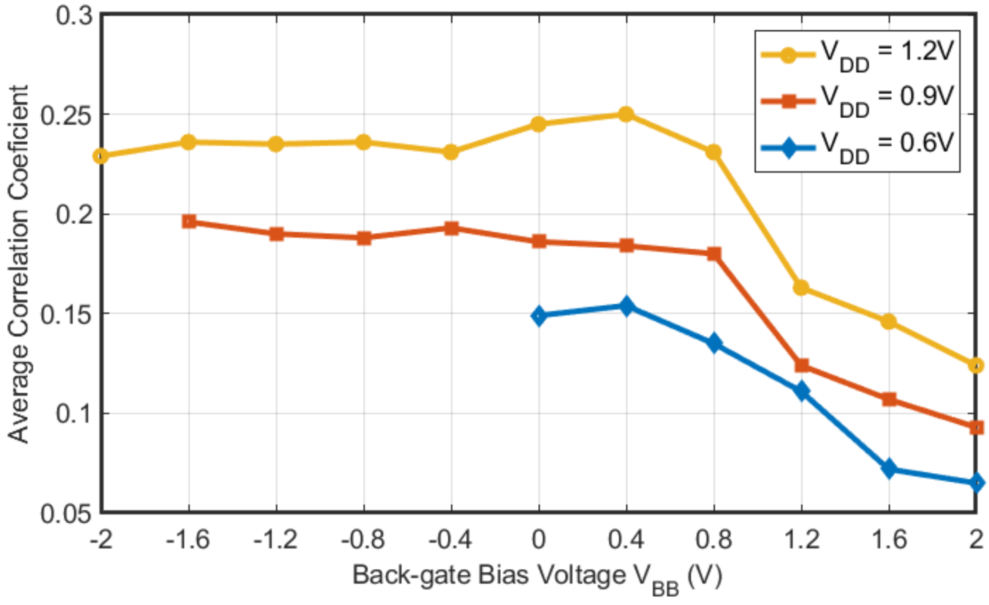


Figure 4.9: DPA Attack results (average of 16 correct subkey's correlation coefficients) at different combinations of fixed V_{BB} and V_{DD}.

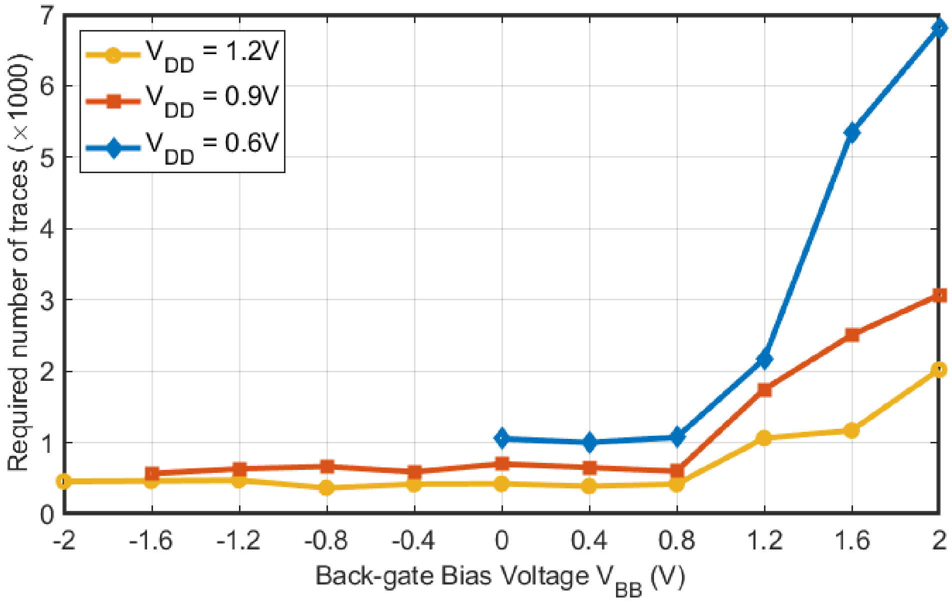


Figure 4.10: DPA Attack results (average of 16 correct subkey's MTD) at different combinations of fixed V_{BB} and V_{DD}.

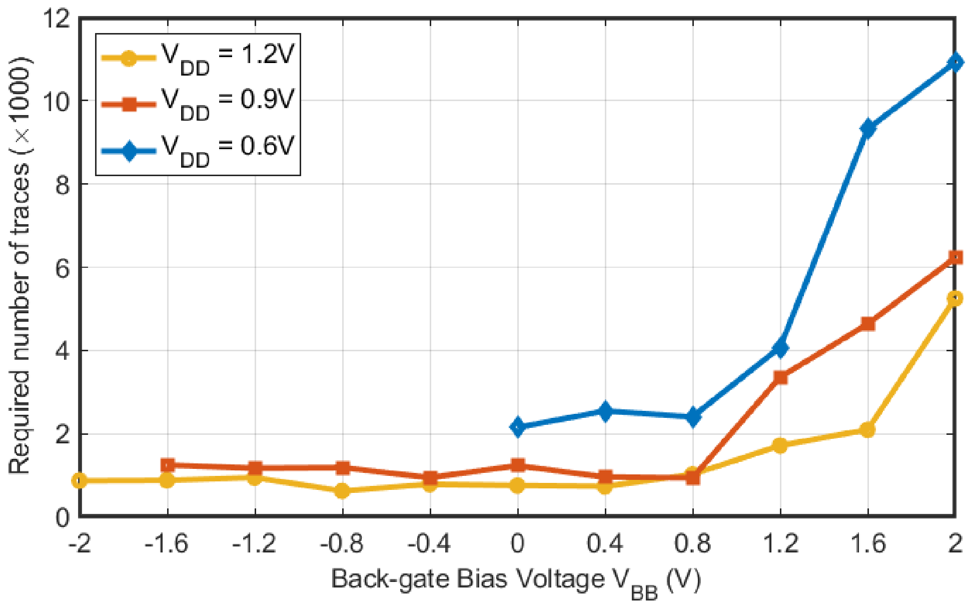


Figure 4.11: DPA Attack results (correct full-key's MTD) at different combinations of fixed V_{BB} and V_{DD}.

consumption and separate it from overall power consumption of an actual circuit. However, the idle power of the circuit could be monitored and utilized to estimate the proportion of power deemed as noise in DPA attacks.

As demonstrated in (4.5), the total power consumption of the microcontroller can be described as the sum of various components.

$$P_{total} = P_{op} + P_{data} + P_{noise} \tag{4.5}$$

where:

P_{total} denotes the total power consumption.

P_{op} denotes the proportion of power that depends on the microcontroller operation.

P_{data} denotes the proportion of power that corresponds to processing data.

P_{noise} denotes the proportion of power contributed by the leakage current, electronic interference, and all other sources of noise.

When the microcontroller is in idle mode, which means that the supply voltage and clock are on but the firmware is halted, no operations or data are performed, the P_{op} and P_{total} can be removed. As a result, P_{noise} can be calculated by monitoring the microcontroller's average power usage in idle mode. The sum of P_{op} and P_{data} is the exploitable leakage power utilized by attackers, represented as P_{exp}, and the ratio of exploitable power to overall power consumption may be determined using (4.6).

$$\frac{100\times(P_{total}-P_{idle})}{P_{total}}\% \tag{4.6}$$

Figure 4.12 shows how varying back-gate biases and supply voltages affect the exploitable leakage power ratio on the targeted microcontroller. Surprisingly, even though the leakage current is lowered, the exploitable power ratio remains constant at around 14.5 percent for the whole reverse back-gate bias region. Furthermore, in the forward back-gate bias region, this ratio drops to less than 1 percent. The rate of the dropping ratio is sharpest between 0.4V and 1.2V V_{BB} and lowers when $V_{BB}\geq$ 1.2V. Examining the exploitable leakage power ratio under various supply voltage and back-gate bias setups could help explain earlier DPA experiment results.

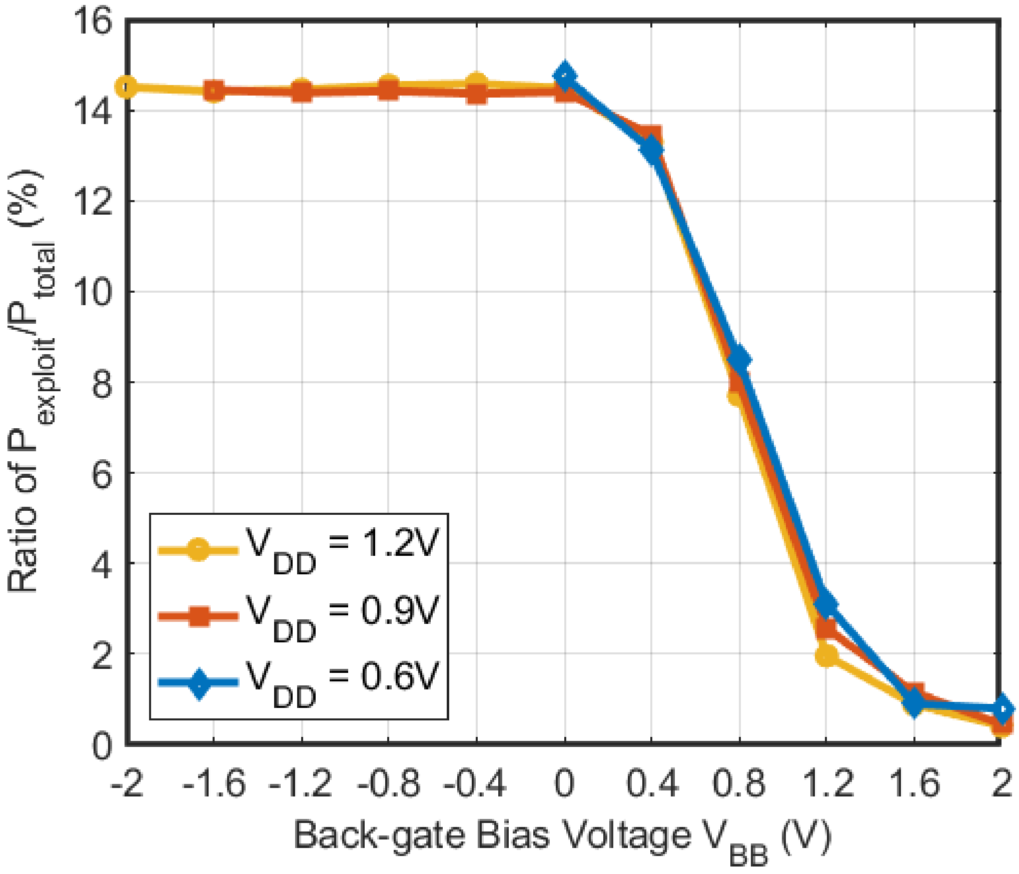


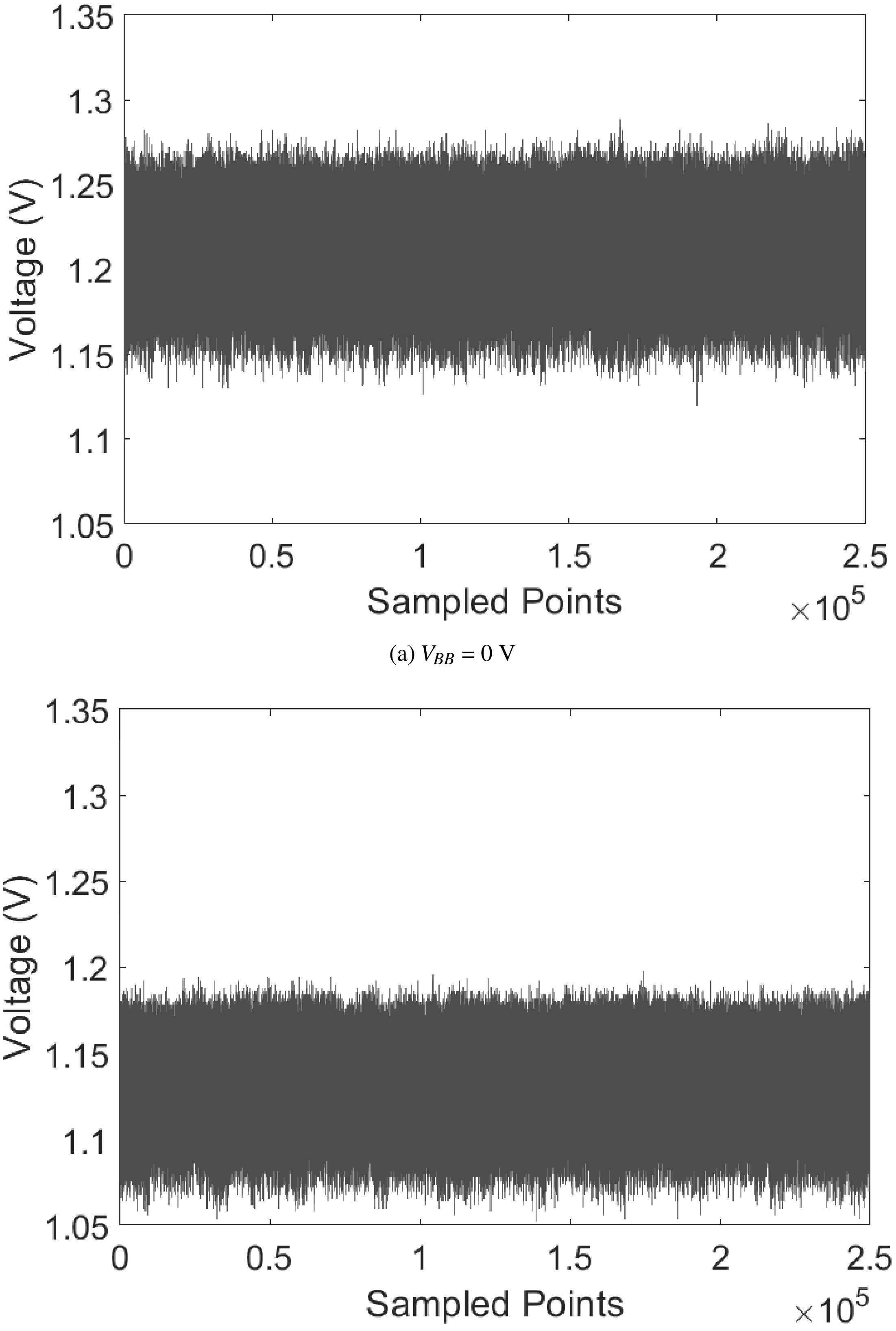
Figure 4.12: Proportion of exploitable power in total power at different combinations of fixed V_{BB} and V_{DD}.

4.3.2.2 Random dynamic back-gate biasing

The previous subsection presented experimental results demonstrating that adjustments in the bias voltage cause nonlinear changes in the exploitable-power-to-total-power-consumption ratio. Furthermore, an increase in total power consumption causes a decrease in power traces measured at the V_{DD} node, and vice versa. Figure 4.13, for example, depicts two measured power traces of the targeted microcontroller configured with no bias voltage in Figure 4.13a and a deep forward bias voltage of V_{BB} = 2V in Figure 4.13b. The power trace measured at a V_{BB} of 2V has a voltage level reduction of approximately 60 mV when compared to the power trace measured at no bias voltage. In general, the randomized bias voltage will result in randomized amplitude misalignments among the acquired power traces. Hence, if the targeted microcontroller executes each encryption/decryption at a random bias voltage voltage, the variation in signal-to-noise ratio and amplitude misalignment of each power trace in the measured data set will also be randomized, thus weakening the correlation between intermediate values and actual power consumption.

The amplitude misalignment of the power traces measured at different back-gate biases and supply voltages must be examined. For each configuration that has a supply voltage of 0.6 V, 0.9 V, or 1.2 V, and a back-gate bias in the range of -2.0V to 2.0V with a 0.1V step, 200 power traces are measured. The means of their minimum and maximum sample point voltage amplitudes are calculated and shown in Figure 4.14. Mean sample points of the same supply voltage configurations are connected to construct line graphs. These line graphs illustrate the amplitude misalignment of measured power traces caused by tuning the back-gate bias. Figure 4.14 and Figure 4.12 can be used to determine the best range for randomizing back-gate bias by identifying the continuous range that causes the most considerable reduction in the voltage level of power traces. In addition, the microcontroller is also required to operate stably under dynamic changing of the back-gate bias. This requirement draws some limitations in selecting the range of dynamic randomizing back-gate bias. When testing on our power trace measuring system, the stability of the targeted microcontroller can be ensured, and a suitable randomizing back-gate bias range can be determined.

On the targeted microcontroller, we carried out a number of DPA attacks with various ranges of the dynamic randomizing back-gate bias. Figure 4.14 shows the optimal range



(a) V_{BB} = 0 V

(b) V_{BB} = 2 V

Figure 4.13: Example of power traces at 1.2V of V_{DD} with 0V ofV_{BB} (a) and 2V of V_{BB} (b).

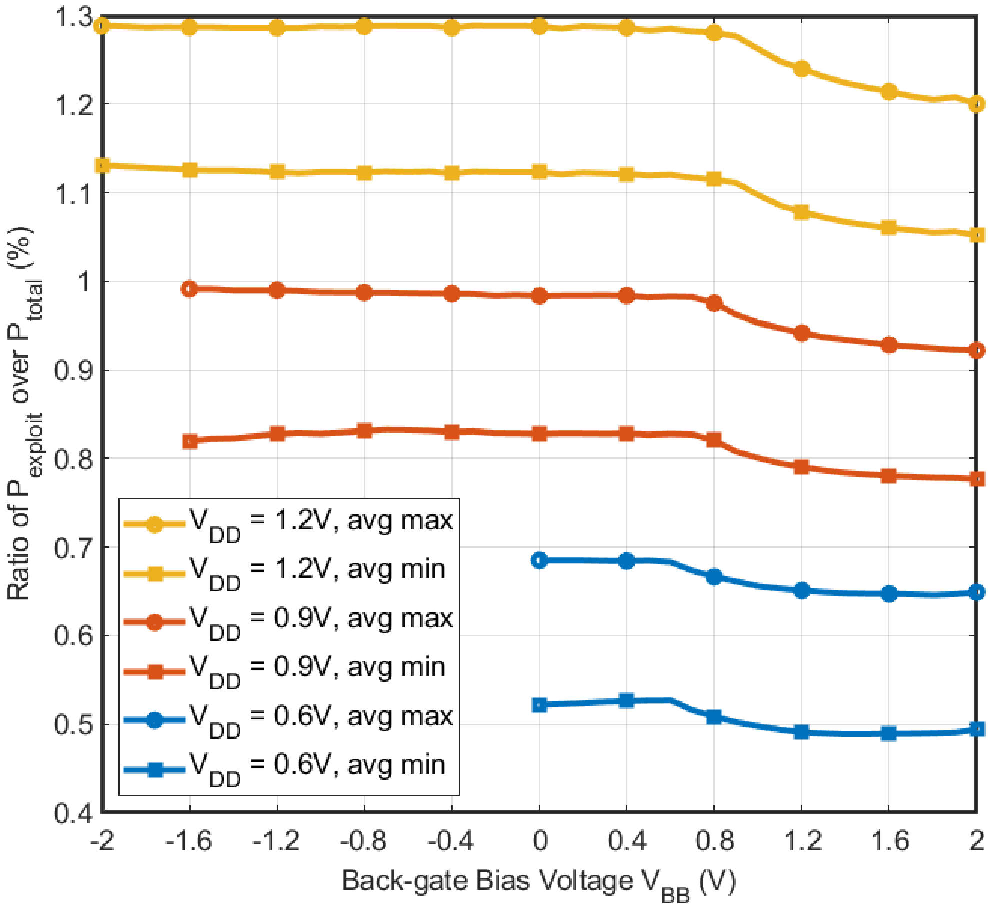


Figure 4.14: Power trace's amplitude variation when different fixed V_{DD} and V_{BB} are applied.

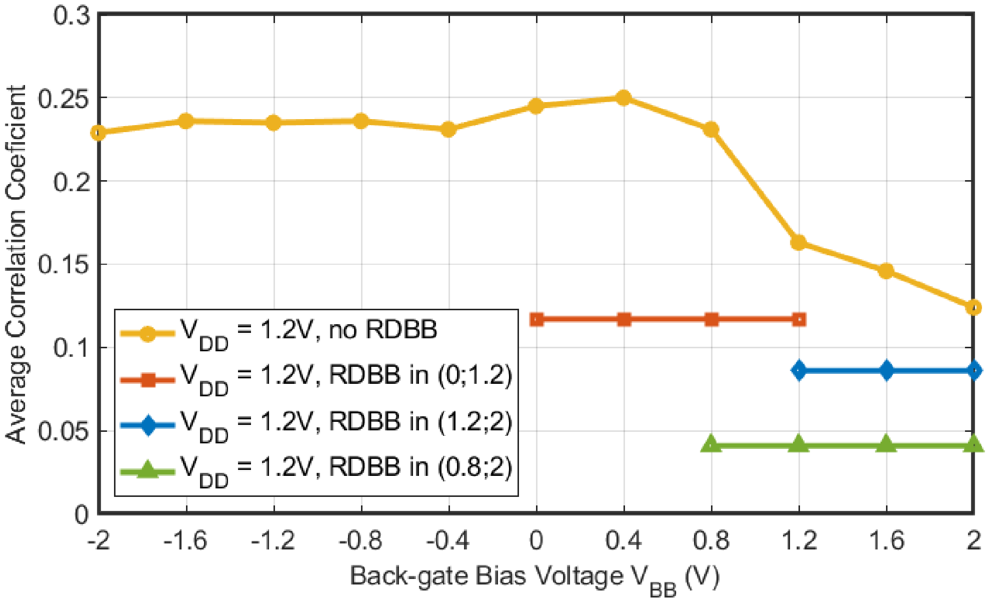


Figure 4.15: DPA Attack results (average of 16 correct subkey's correlation coefficients) at 1.2V of V_{DD}, with several RDBB configurations.

of back-gate bias V_{BB} for each specificed supply voltage V_{DD}. The optimal range of the dynamic randomizing back-gate bias is from 0.8V to 2V when the supply voltage V_{DD} = 1.2V. Even with this RDBB configuration, the targeted microcontroller remains stable while the measured power traces experience the greatest voltage variation. The highest sample point of the measure trace reduces by 81 mV, while the minimum sample point decreases by 63 mV, as shown in Figure 4.14. Figure 4.15, Figure 4.16, Figure 4.17 show the effectiveness of using RDBB when the supply voltage V_{DD} = 1.2V. As expected, the best DPA resistance is obtained when the RDBB is configured between 0.8V and 2V. The average of the correlation coefficients of the correct hypothesis is only 0.041 with this configuration. This is a reduction of almost six times when compared to the reference DPA attack result with no bias and a 1.2V of supply voltage V_{DD}. The average number of traces necessary to recover a subkey increases by 31.3 times, from 422 to 13,238, while the least number of traces required to recover a full 16-bytes secret key increases by 33 times, from 755 to 24,927. For better evaluation, the DPA attack results when RDBB is applied in other stable ranges are also provided (from 0V to 1.2V and from 1.2V to 2 V). The results suggest that using RDBB in other ranges can also raise DPA resistance of the target microcontroller, however the efficiency cannot be compared to that of the 0.8 to 2V range.

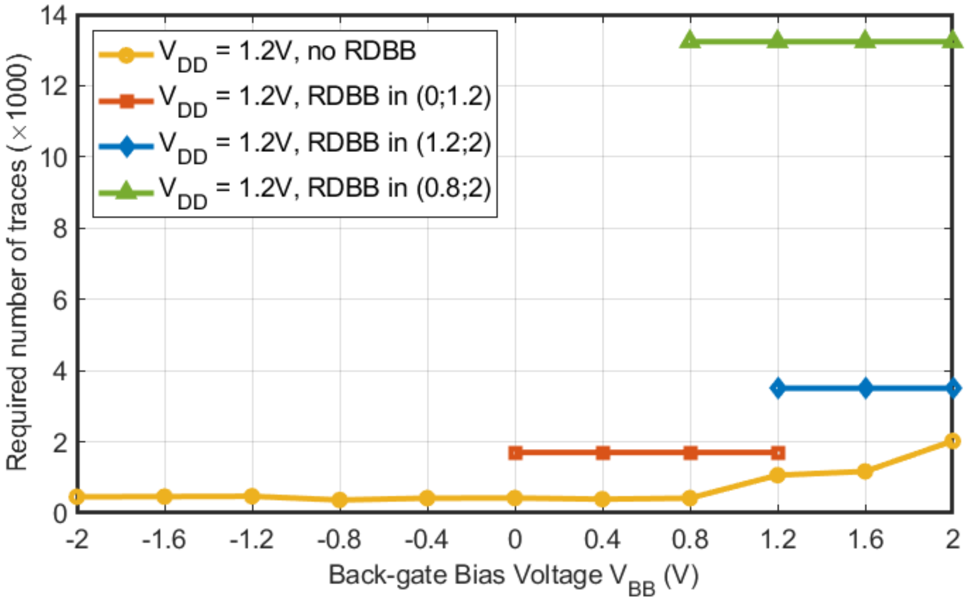


Figure 4.16: DPA Attack results (average of 16 correct subkey's MTD) at 1.2V of V_{DD}, with several RDBB configurations.

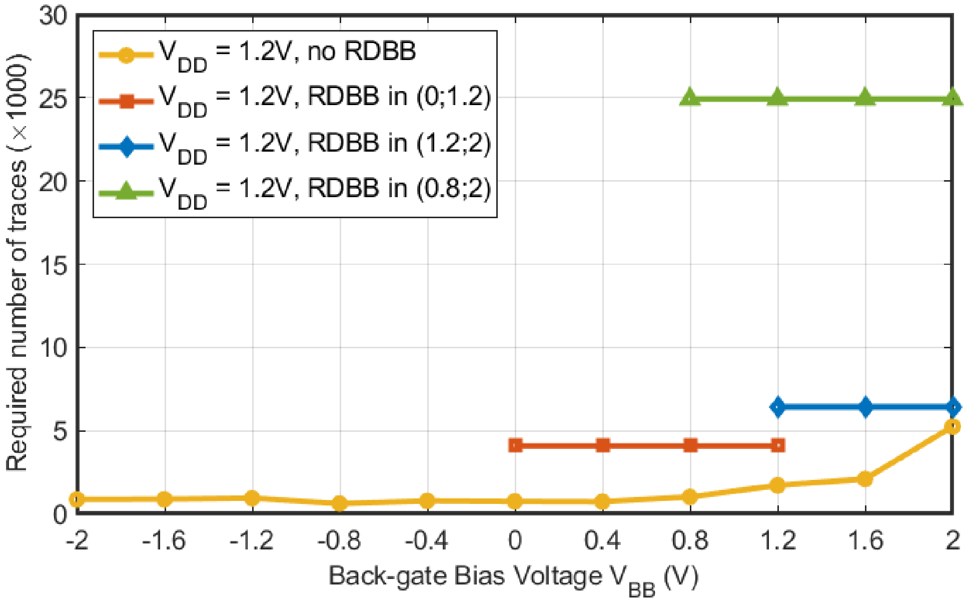


Figure 4.17: DPA Attack results (correct full-key's MTD) at 1.2V of V_{DD}, with several RDBB configurations.

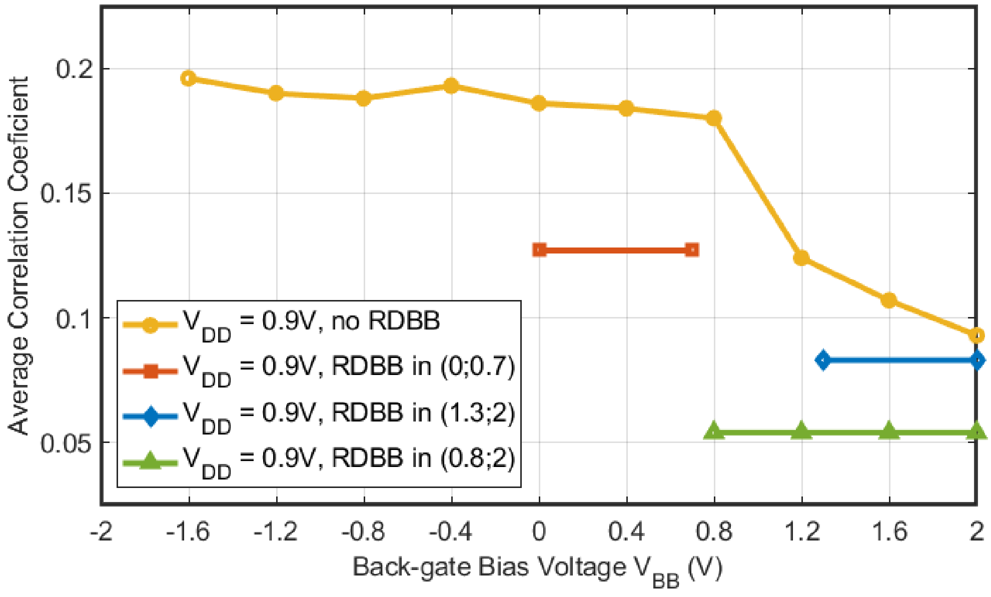


Figure 4.18: DPA Attack results (average of 16 correct subkey's correlation coefficients) at 0.9V of V_{DD}, with several RDBB configurations.

Likewise, scenarios with supply voltages of 0.9V and 0.6V are tested. When the supply voltage V_{DD} is 0.9 V, the most stable range of dynamic randomizing is from 0.8V to 2 V, and from 0.6V to 1.4V when the supply voltage V_{DD} is 0.6V. The results of related DPA attacks are displayed in Figure 4.18, Figure 4.19, Figure 4.20, Figure 4.21, Figure 4.22, and Figure 4.23. The average of the correlation coefficients of the correct hypothesis is decreased to only 0.054 in Figure 4.18, which is 3.4 times less than the DPA attack result with no bias and a 0.9V V_{DD}. Furthermore, the average number of traces needed to recover a subkey increases by 13.44 times, from 699 to 9,401, and the least number of traces needed to recover a whole secret key increases by 13.21 times, from 1,230 to 16,249. The average of the correlation coefficients of the correct hypothesis is decreased to 0.048 in Figure 4.21, which is 3.1 times less than the DPA attack result with no bias and a 0.6V V_{DD}. Moreover, the average number of traces needed to recover a subkey increases by 10.4 times, from 1,052 to 10,942, and the least number of traces needed to recover the whole secret key increases by 9.43 times, from 2,150 to 20,275. At lower supply voltages, the effectiveness of the RDBB countermeasure is reduced. These DPA attack results are consistent with the assessment of amplitude misalignment among recorded power traces illustrated in Figure 4.14, where power traces measured with a lower supply voltage have smaller voltage reduction.

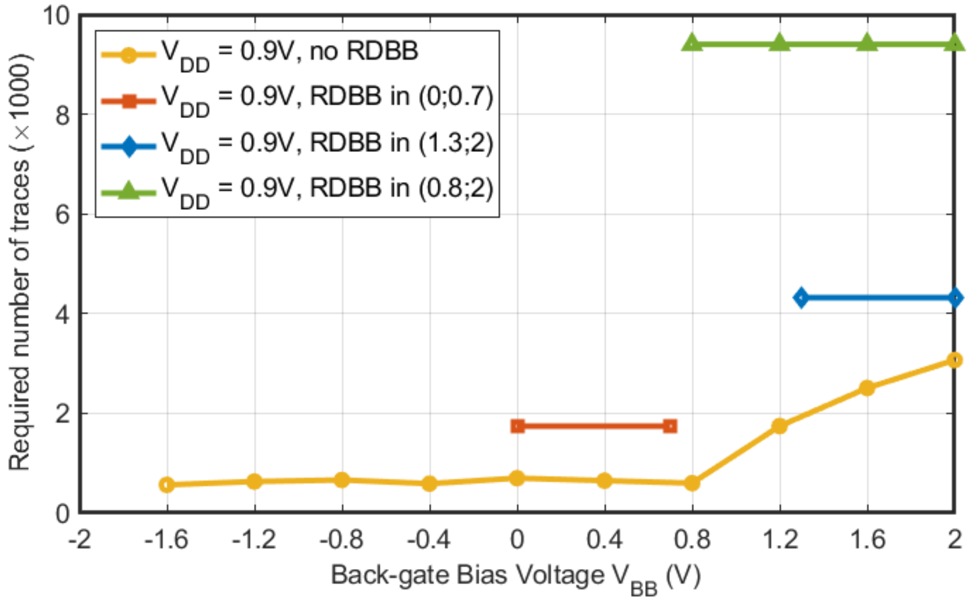


Figure 4.19: DPA Attack results (average of 16 correct subkey's MTD) at 0.9V of V_{DD}, with several RDBB configurations.

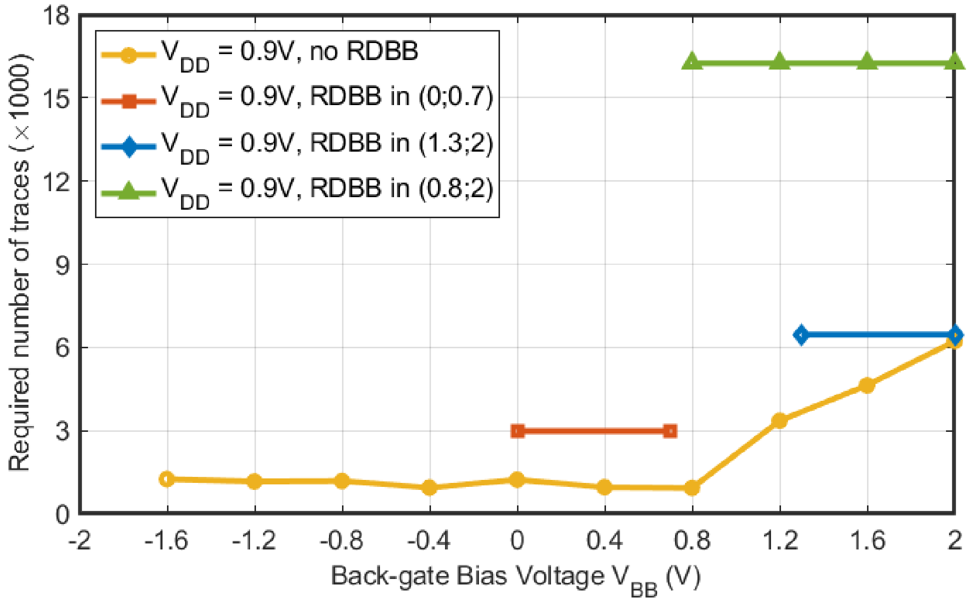


Figure 4.20: DPA Attack results (correct full-key's MTD) at 0.9V of V_{DD}, with several RDBB configurations.

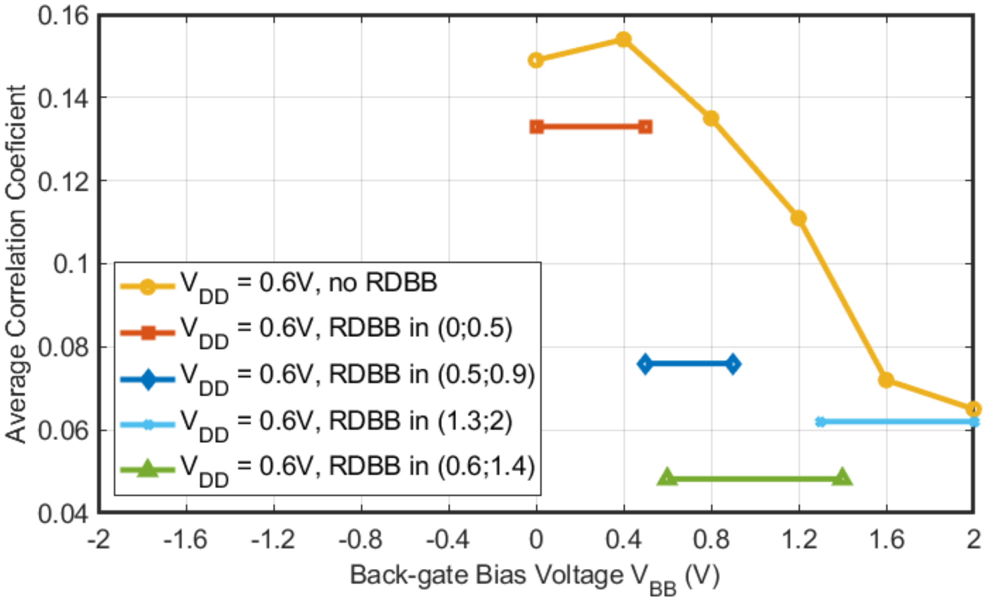


Figure 4.21: DPA Attack results (average of 16 correct subkey's correlation coefficients) at 0.6V of V_{DD}, with several RDBB configurations.

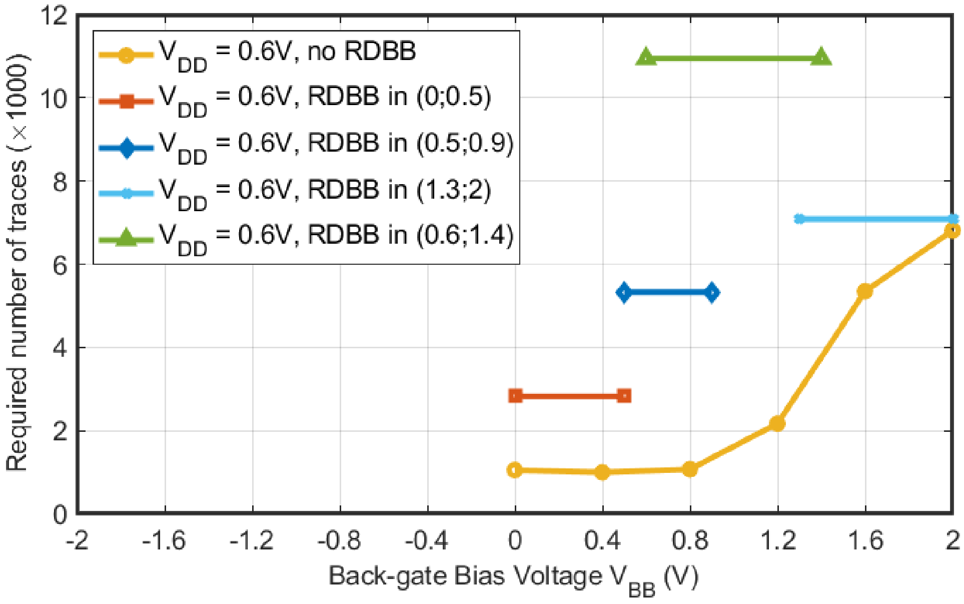


Figure 4.22: DPA Attack results (average of 16 correct subkey's MTD) at 0.6V of V_{DD}, with several RDBB configurations.

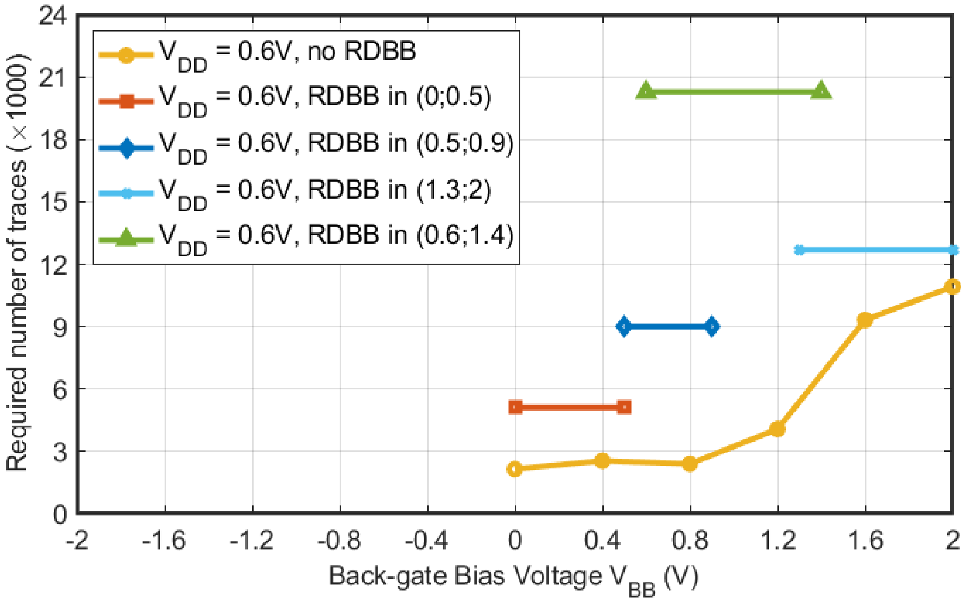


Figure 4.23: DPA Attack results (correct full-key's MTD) at 0.6V of V_{DD}, with several RDBB configurations.

4.3.3 Result comparison

When comparing the effectiveness of different DPA countermeasures, DPA attack results conducted on comparable devices with comparable cryptographic implementations should be used. Two novel DPA hiding countermeasures are presented and demonstrated on a microcontroller with AES-128 implementation in this study. Thus, in order to further assess the efficiency of the proposed countermeasures, the experimental results are compared to those of other hiding countermeasures evaluated using comparable assessment methodologies. Table 4.1 compares our proposed countermeasures to alternative hiding countermeasures, which are tested using practical DPA attacks on comparable devices. Because the forward back-gate bias and low supply voltage are applied directly utilizing off-chip components, the first proposed method (using fixed forward bak-gate bias) increased the number of required traces by 14.5 times while causing no additional power, space, or performance overhead. Our second proposed method increased the required number of traces by 33.4 times and was also demonstrated using off-chip components. In practice, however, these countermeasures require the use of an on-chip random number generator and an on-chip voltage regulator to provide the random back-gate bias voltage. This practical requirement may result in a minor increase in power, area, and performance. In this study, the proposed countermeasures are tested on a manufactured microcontroller that lacks an on-chip random number generator and a voltage regulator. Therefore, the specific overhead metrics induced by the second proposed approach are unavailable and are shown as N/A in Table 4.1.

First, we compare our proposed countermeasures to the RDVS countermeasure [47]. The original evaluation results provided in [47] are just simulation results. Therefore, for the effectiveness comparison, the RDVS countermeasure is also applied to the target 32-bit RISC-V microcontroller. Similar to RDBB, the dynamic randomizing supply voltage range must be determined. As stated in [47], simulation findings on the AES S-Box show that raising the randomizing range of the supply voltage reduces correlation strength. Therefore, [47] proposed to increase the maximum or decrease the lowest supply voltage in their simulation. Unfortunately, when targeting a real microcontroller, the higher and lower supply voltage boundaries are severely limited to preserve system stability. The maximum stable range of the dynamic randomizing supply voltage after testing on the automated power trace measuring system, is determined from 0.9V to 1.2V. The targeted microcontroller can operate with all back-gate biases in the forward bias region after applying this range of RDVS. Figure 4.24, Figure 4.25, and Figure 4.26 show the DPA attack results on the targeted microcontroller with RDVS applied at various different back-gate biases. It demonstrates that RDVS with a supply voltage takes values ranging from 0.9V to 1.2 V, resulting in increasing DPA resistance, but this action is only as effective as lowering the supply voltage of the targeted microcontroller to 0.6V. The line graphs of both cases overlap and are nearly identical. If no bias voltage is used, 2,572 traces are required to reveal the entire 16 bytes of the secret key. Meanwhile, If a deep forward bias voltage of 2V is applied, that number rises to 11,456 traces. As a result, the number of necessary traces grows by 15.2 times in the best scenario, when RDVS and a deep forward bias of 2V are used. If no bias voltage is utilized, the improvement is only 3.4 times. Furthermore, the RDVS countermeasures are implemented by utilizing off-chip components. However, similar to the proposed RDBB countermeasure, the RDVS countermeasure also requires to use an on-chip random number generator and an on-chip voltage regulator for generating random supply voltages. Therefore, it can be concluded that the second proposed RDBB countermeasure is more effective than the RDVS countermeasure while requiring the same additional overhead. The specific overhead metrics

Table 4.1: Comparison with other countermeasures

	Countermeasure Used	Test Device	Overhead			Increase in number of required traces
			Power	Area	Performance	
This work	Forward Back-gate Bias and Low Power Supply Voltage (proposed)	32-bit RISC-V MCU 65-nm SOTB [41]	0			14.5×
	Random Back-gate Bias (proposed)		N/A			33.4×
	Random Dynamic Voltage Scaling[47]		N/A			15.2×
TVLSI'20[51]	Random Task Scheduling	Multicore Processor TSMC 65-nm CMOS LP	3.5%	2.3%	4%	7× (2.75×)[1]
	Random Insertion of Operations					280× (3×)[1]
	Frequency and Phase Randomization					2×
	Power State Monitoring and Control					45×
ISSCC'17[31]	Integrated Voltage Regulator	AES Core 130-nm CMOS	5%	1%[2]	3.33	100×
TCAS-I'18[46]	Attenuated Signature Noise Injection	AES Core 130-nm CMOS	68%	60%	0	> 1000×
TVLSI'18[69]	Secure Double Rate Registers	AES Core 65-nm CMOS	180%	33%	0	> 1388×
ISSCC'19[48]	Switching Noise Injector and Randomized Reference Voltage	AES Core 130-nm CMOS	32%	36.90%	10.40%	> 3579×

[1] The effectiveness is significantly decreased by aligning the traces.

[2] Only the loop randomization block is counted as area overhead, not the entire voltage regulator.

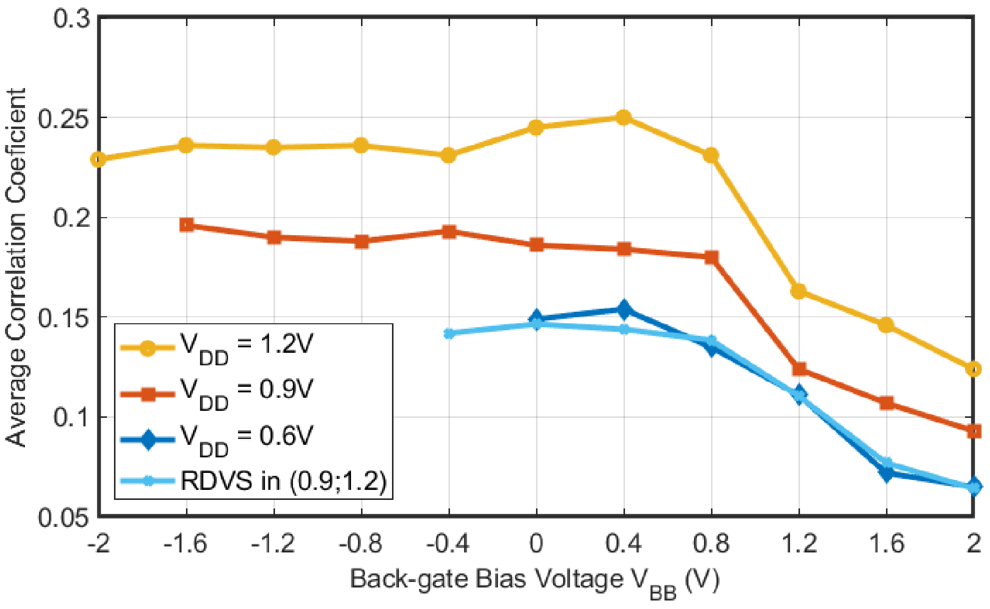


Figure 4.24: PA Attack results (average of 16 correct subkey's correlation coefficients) with different fixed V_{BB} and RDVS countermeasure.

induced by the RDVS countermeasure are also displayed in Table 4.1 as not available (N/A).

A multicore processor equipped with randomization hiding countermeasures was reported in by Yang *et al.* [51]. They integrated several countermeasures into a multicore processor to enhance its DPA resistance, including random task scheduling (RTS), random insertions of operation (RIO), frequency and phase randomization (FPR), and power state monitoring control (PSMC). Each of these techniques raises the number of required traces to reveal a secret subkey by 7, 280, 2, and 45 times, respectively. However, the RTS and RIO countermeasures obtain these results by misaligning the power trace only in the time axis. In[51], authors also proved that the misalignment in the time axis can be eliminated by using correlation-based realigning techniques. When realignment methods are used, the RTS and RIO countermeasures can only raise the number of required traces to reveal a secret subkey by 2.75 and 3 times, respectively. Our proposed DPA resistance improvement is accomplished by adjusting the noise level in power traces, or in other words, by introducing noise into the amplitude dimension. Thus, our proposed countermeasures are not restricted by existing realignment approaches.

The power overhead, area overhead, and performance overhead reported in are 3.5 percent, 2.3 percent, and 4 percent, respectively. The additional phase-locked loop (PLL),

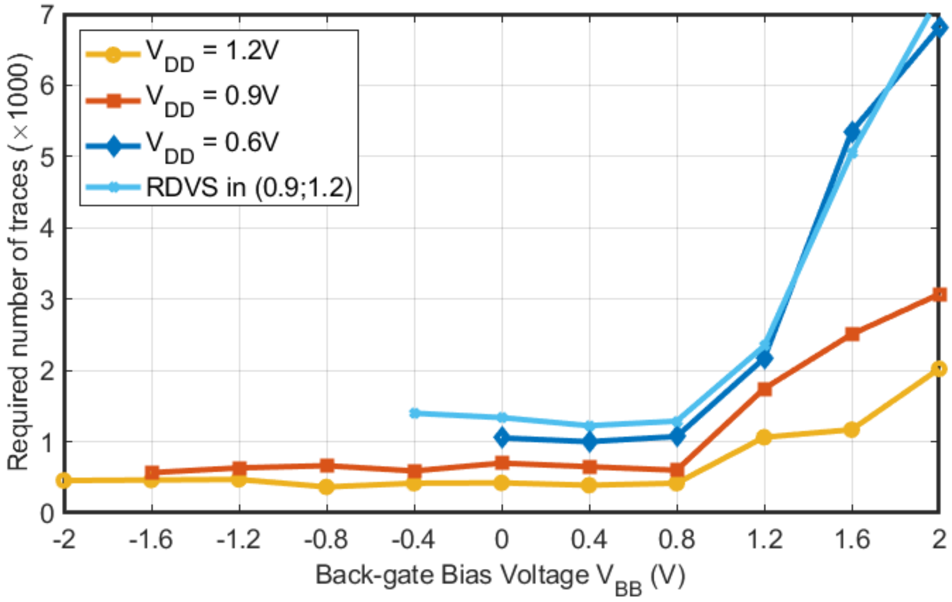


Figure 4.25: DPA Attack results (average of 16 correct subkey's MTD) with different fixed V_{BB} and RDVS countermeasure.

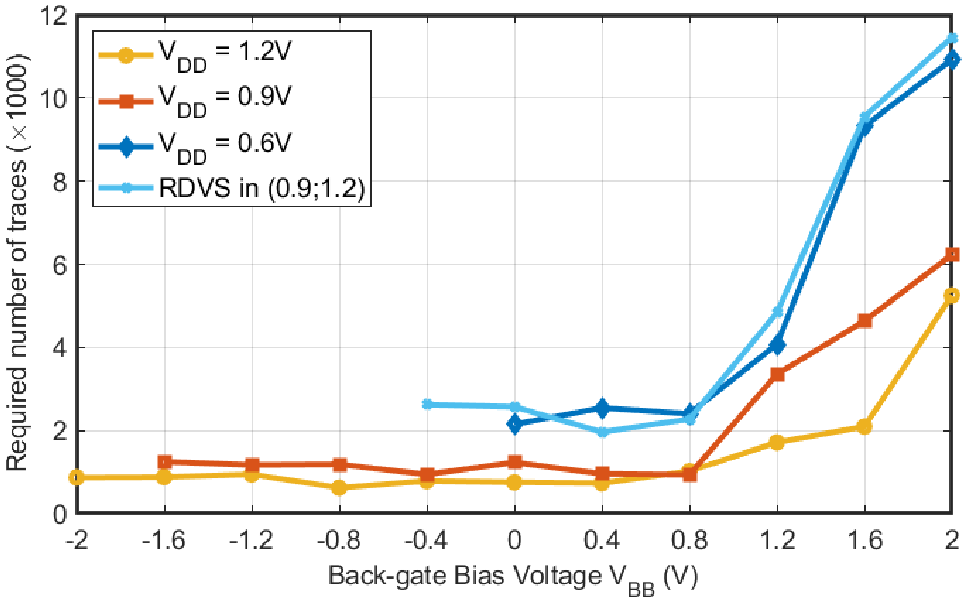


Figure 4.26: DPA Attack results (correct full-key's MTD) with different fixed V_{BB} and RDVS countermeasure.

stall controller, and randomization controller are responsible for these overall overheads. The randomization controller is used by all four countermeasures. Only the PSMC countermeasure employs the stall controller, meanwhile only the FPR countermeasure employs the PLL. Yang *et al.* does not publish the particular power, area, and performance of these additional hardware components. Therefore, it is impossible to quantize the power, area, and performance overheads of each specific countermeasure. We can only provide an approximate comparison of the power, area, and performance overheads of our proposed countermeasures with those employed in [51]. As can be observed in Table, our first proposed idea offers significantly better DPA resistance than the RTS, RIO, and FPR countermeasures while requiring no additional hardware. At the estimated cost of an additional on-chip voltage regulator, the second proposed RDBB countermeasure also delivers stronger DPA resistance than the RTS, RIO, and FPR countermeasures. In [51], the randomization controller includes a frequency controller, a phase controller, and a true random number generator. In comparison to [51], our proposed RDBB countermeasure also needs the use of an on-chip random number generator. However, no phase-locked loop (PLL), stall controller, eight phase-controllers, or frequency-controller are required. Using our proposed RDBB countermeasure also necessitates the use of an on-chip voltage regulator. Even though the individual overheads generated by the additional on-chip voltage regulator differ depending on specific designs, they are estimated to be negligible, as indicated in [31]. As a result, the total overhead metrics associated with our second proposed approach are predicted to be comparable to those published in [51], which are only a few percent. Furthermore, the RTS and PSMC countermeasures are only applicable to multicore computers since they require arbitrarily changing and switching processes across different cores. Hence, the RTS and PSMC countermeasure cannot be adapted to diverse designs, such as single-core processors and cryptographic hardware accelerators, as opposed to our proposed countermeasures using back-gate bias.

DPA-resistant AES cores with various hiding countermeasures were introduced in [31, 46, 69, 48]. In terms of the number of traces required to reveal the secret key, these countermeasures give a significant enhancement. Unfortunately, they require significantly more power, area, and performance overheads. Therefore, unlike our proposed countermeasures, these countermeasures are only suitable for small designs.

We also attempted to combine RDVS and RDBB by randomly varying both the supply voltage and the back-gate bias. Unfortunately, when both RDVS and RDBB countermeasures are deployed simultaneously, the voltage range of supply voltage and back-gate bias, that allow stable operation of the targeted microcontroller, is significantly decreased. Consequently, the DPA resistance of the targeted device is not as excellent as when RDBB is used alone. The best results are obtained when RDVS with a supply voltage ranges from 0.9V to 1.2V and RDBB back-gate bias ranges from 0.8V to 1.6V. In that configuration, the least number of traces required to recover all 16 bytes of the secret key in that configuration is 12506 traces, which is approximately 16.5 times greater than the reference configuration (supply voltage of 1.2V, no back-gate bias, and no countermeasures applied).

4.4 Further Evaluations

4.4.1 Evaluation with average-measuring technique

The approach of the countermeasure proposed in this chapter is based on increasing the noise level in power consumption traces. Therefore, a smart attacker would apply certain techniques to diminish the effects of the proposed countermeasure. For example, a simple and effective technique is the average-measuring technique. The encryption on the same plaintext are performed several times. All power traces related to these encryption are considered as raw power traces with same length. The sample-by-sample average trace of these raw power traces are computed and then used as final trace for later power analysis attacks. In this section, the proposed countermeasure of exploiting the back-gate bias is further evaluated with the practical experiments where the average-measuring technique is applied by the attackers.

4.4.1.1 Experimental setup

Various additional experiments of power analysis attacks with utilization of average-measuring technique on the target RISC-V MCU will be conducted. The target MCU will operate with different configuration of back-gate bias voltage V_{BB} and supply voltage V_{DD}. After that, these experimental results will be compared with that of experiments described in section 4.3.2. For demonstration purpose and without the loss of generality, in this section, only some configurations of V_{DD} and V_{BB} are chose to be experimented. The selected configurations are:

- Fixed V_{DD} of 1.2V and fixed V_{BB} in the range from 0V to 2.0V with 0.4V increment step.
- Fixed V_{DD} of 1.2V and random dynamic back-gate bias in the range from 1.2V to 2.0V

The automatic power trace measuring system presented in section 4.3.1 is used to measure the power traces for these various additional experiments. The measuring system are slightly modified for using the average-measuring technique. The Oscilloscope's acquisition mode is changed to average-8 mode, in which 8 raw traces are measured to produce a final averaged trace. The Control PC would repeatedly send the plaintext to the target MCU 8 times, so that 8 encryption on the same plaintext are performed. The trigger signal via GPIO is still toggled at the start of each encryption. When the RDBB countermeasure is applied, the Back-gate Bias is still changed after each encryption. The Oscilloscope also supports other average-measuring mode such as average-2, average-4, average-16, average-32, average-64 with corresponding 2, 4, 16, 32, 64 raw traces before final trace computation. More raw traces means less random noise in the final trace but more acquisition time are required. The average-8 mode is chose to maintain a reasonable acquisition time. The final averaged traces are analyzed by using the same MATLAB application described in subsection 4.3.1.3.

4.4.1.2 Experimental results

The experimental attack results are presented in Figure 4.27 and Figure 4.28. In all additional experiments, all 16 bytes of secret key are quickly and correctly revealed. Figure 4.27 shows the average correlation coefficient of revealed subkey. It can be seen that using the average-measuring technique help to minimize the random noise in the final averaged power traces, hence improve the correlation coefficients of the correct subkey hypothesis, making it easier to be distinguished with other incorrect subkey. As results, Figure 4.28 shows that using average measuring technique helps to significantly reduce the number of required traces to recover the secret key. In the best scenario, when the Back-gate bias is randomized in the range from 1.2V to 2V, the average-8 measuring reduce the number of required traces from around 3500 traces down to nearly 500 traces. However, these 500 traces are final averaged traces. They are equivalent with 4000 raw power traces since we are using the average-8 mode. It means that the average measuring technique is not really effective in term of measuring time.

Figure 4.27 and Figure 4.28 also indicate that even when the attacker uses the average-measuring technique, the proposed countermeasure still has the same effects. Biasing a deep forward bias voltage still helps to improve target device's resistance against power analysis attacks. The resistance is further improved when the back-gate bias is randomized in the forward bias region, after each encryption. This experimental results agree with a recently published article [70]. In [70], authors studied the effect of average-measuring technique on the Random Body Bias countermeasure for FD-SOI devices. They provided theoretical analysis and numerical simulation to show that the behavior of the Random Body Bias countermeasure is similar in all cases, regardless of the number of raw traces used for averaging.

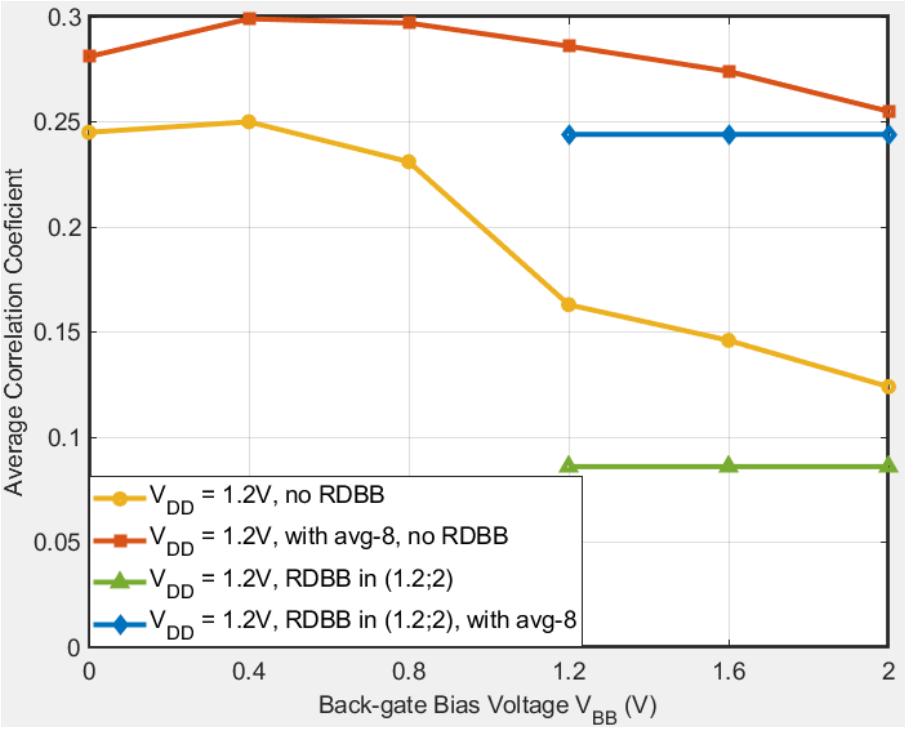


Figure 4.27: Average correlation coefficient of the correct subkey in additional experimental results.

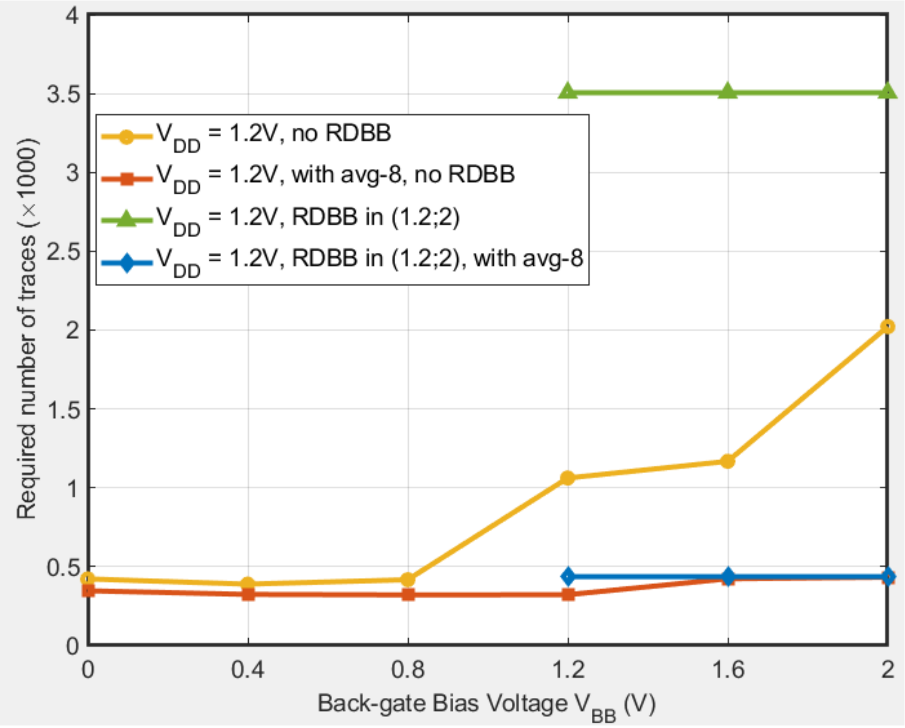


Figure 4.28: Average required number of trace to reveal the correct subkey in additional experimental results.

4.5 Summary

This chapter offered two innovative countermeasures to DPA assaults that make use of the FD-SOI technology's back-gate bias approach. The first proposal was to bias the targeted device in the forward bias region to lower the ratio of exploitable power helpful to attackers to total power consumption. As a result, the back-gate bias technique not only improves integrated circuit performance but also aids in the defense against DPA attacks. The second idea was a novel random dynamic back-gate biasing countermeasure, which involves dynamically randomizing the targeted device's back-gate bias in order to increase its resistance to DPA attacks. The effectiveness of our proposals was assessed by performing actual DPA attacks on a 32-bit RISC-V Microcontroller built using SOTB technology. According to the experimental results, our first proposed countermeasure can increase the number of required traces to correctly recover entire 16-byte secret key used in AES-128 by 14.5 times. Furthermore, the second proposed RDBB countermeasure has the potential to boost that figure up to 33.4 times. This result is more than twice as good as the previously reported RDVS countermeasure [47], since practical DPA attacks on the same targeted device reveal that the RDVS countermeasure can only increase the amount of required traces by 15.2 times in the best scenario.

Chapter 5

Dynamic Frequency Scaling Countermeasures

5.1 Proposed Idea

Using heterogeneous System-on-Chips (SoC) is now the most popular way to construct embedded systems with high-performance. VLSI advancements enable the integration of all major components of a traditional computer into a System-on-Chip. The major components include numerous memory blocks, a number of peripheral circuits or hardware accelerators, and one or many processing cores. An application-specific processor, a microprocessor, or microcontroller a can be used as the processing cores. The processing core operates the software programs stored in memory space. High-level users can reprogram, tweak, and improve the software to match with any general purpose. Peripheral circuits and hardware accelerators, on the other hand, give more advanced performance for some intensive, particular applications. Data collection, Tensor processing, and cryptographic computation are some examples of these tasks. Therefore, the SoC architecture enables diverse and dependable solutions for a broad range of applications by combining the flexibility of programmable software with the performance of bespoke hardware accelerators.

The RISC-V Foundation recently released the open licensed RISC-V instruction set architecture (ISA). This ISA has been an attracting topic for various SoC designers in academia and industry since it eliminates the majority of the restrictions related to working with proprietary SoC's components and design tools. The SoC designers could effortlessly develop their customized RISC-V SoCs to meet their specific requirements. Within a few years, the RISC-V SoC has gained widespread adoption and use in a variety of fields. Among all of these many uses, security is gaining significant traction. Numerous researchers and designers recognized that the RISC-V SoC's accessibility and flexibility may be leveraged to execute their ideas for increasing secure embedded systems' reliability and performance. As a result, RISC-V SoCs are integrated with cryptographic accelerators for a variety of applications. Duran *et al.* proposed a RISC-V SoC that has an upgraded memory accessing mechanism and capable of executing customized instructions to speed up the AES-256 algorithm [71]. Zang *et al.* proposed an system based on a RISC-V processing core combined with an AES-128 accelerator for secure communication in Internet-of-Thing usage [72]. For instance, in [73], Banerjee *et al.* attached an reconfigurable and energy-efficient cryptographic module into a RISC-V SoC to apply in Datagram Transport Layer Security (DTLS) applications. The cryptographic module comprises several hardware accelerators, including Secure Hash Algorithm computations (SHA), Advanced Encryption Standard (AES), and Elliptic Curve Cryptography (ECC).

Meanwhile, power analysis attacks has became remarkably popular and effective. Numerous studies have showed that these type of attacks can easily break of various cryptographic implementations' security, including dedicated hardware hardware implementations in both ASICs and FPGA technologies [48, 43], as well as software implementations on smart cards or microcontroller [11, 44]. In these broke implementations, the cryptographic process involves most of components in target hardware. Therefore, the power traces measured from these devices contain a small amount of random noise contributed by unrelated modules and a huge amount of leaked information caused by the involved modules. This is in favor of the attackers. On the other hand, breaking the security of complicated systems such as SoCs with multiple cryptographic accelerators is much more difficult because these system are consisted of too many different type of components (such as accelerators, peripheral, bus interconnect system, and processing cores) Beside, all these different type of components are supplied from a single off-chip power supply. Exploitable part of cryptographic accelerators' power traces cannot be separated from the substantial, unexploitable part contributed by other unrelated SoC modules. That means the cryptographic SoC's power traces are noisy or even significantly distorted, posing challenges for attackers.

As a consequence, despite the fact that numerous cryptographic SoCs have been proposed, there is a lack of study addressing the vulnerability of these system to power based side channel attacks. Some example of previously published works include [6, 45]. In [45], Hettwer *et al.* attacked an AES peripheral that implemented on a Xilinx Zynq Ultrascale+ SoC's Programmable Logic part. However, the Xilinx Zynq Ultrascale+ SoC's Programmable Logic part is powered separately with the main processing core [74]. Thus, Hettwer *et al.* may measure and evaluate the electromagnetic traces released by Programmable Logic's decoupling capacitor, which includes only exploitable leakage. Cai *et al.* conduct tests targeting a SoC equipped with a dedicated cryptographic co-processor. That co-processor is used to execute the AES-128 algorithm [6]. The total power usage traces of the SoC are obtained and evaluated to determine the secret key used in AES-128 encryption.

Because of the severe shortage in related works discussing the packed cryptographic SoCs's resistance against PA attacks and conducting decent security evaluations, there are even fewer works that offer solid solution for preventing or countering PA attacks on cryptographic SoCs. Hence, in this chapter, demonstration is firstly provided to confirm that integrated cryptographic SoC are prone to conventional PA attacks in realistic attacking environment. After that, we offer a hiding countermeasure for enhancing the integrated cryptographic SoC's resistance against Power Analysis attacks. The proposed countermeasure is called Random-Dynamic-Frequency-Scaling (RDFS) Its objective is to randomly change only the cryptographic accelerators' clock frequency after each encryption/decryption. Meanwhile, the clock frequency of other hardware components are maintained the same. We illustrate the proposed method's effectiveness by implementing the entire RISC-V SoC on a Sakura-X FPGA board, applying the proposed RDFS countermeasure, and evaluating overall SoC's security. The evaluation is carried out by conducting the Test Vector Leakage Assessment (TVLA), CPA attacks, and the profiled Deep-Leaning based Side-Channel-Analysis (DL-SCA) attacks.

5.2 Related Works

Several previous publications presented a variety of effective hiding countermeasures, all of which are based on the principle of randomizing the operating conditions of the targeted device in order to mitigate power analysis attacks [32, 47, 75]. Yang *et al.* proposed a DPA countermeasure called Random-Dynamic-Voltage-and-Frequency-Scaling (RDVFS), in which the pair of supply voltage-operating frequency is randomly changed during runtime [32]. Subsequently, Baddam *et al.* recommended to change only the supply voltage rather than the pair of supply voltage-operating frequency, claiming that changes in operation frequency are easily noticeable by examining power consumption trace's amplitude peaks [47]. Additionally, last chapter also proposed using the RDBB countermeasure to increase the resistance against power analysis attacks of FD-SOI devices. However, the exploiting back-gate bias countermeasures are applicable only to ASIC-based implementations, not to FPGA-based implementations, as they involve using unique fabrication properties such as working with a large supply voltage range, or regulating the back-gate bias. Recently, multiple publications have reported on the usage of randomizing only the cryptographic circuit's clock signal as a Power Analysis countermeasure [42, 43, 44, 45]. These methods are quite similar, which is using off-chip oscillating source to produce a large number of substituted clocks. These generated clocks are used to drive the cryptographic modules. One significant advantage of this approach is that it does not need any particular fabrication technology feature. As a result, it may be implemented on both ASICs and FPGAs.

Prior works often produce substituted clocks by using the Clock Manager modules and then use clock multiplexers to drive suitable clock signals to the cryptographic circuits. Guneysu *et al.* employ 2 Xilinx FPGA's Digital Clock Managers available to generate 8 same-frequency output clocks that has 8 different phases [42]. A multiplexers network, which is controlled by a randomized selection signal, is used to combine all 8 different phase clock signals into a single output clock signal. That output clock signal is used to drive the cryptographic modules. While this countermeasure efficiently increases resistance against DPA, the pulse rate of the cryptographic modules' driving clock signal is very low, resulting in a considerable reduction in the throughput of a cryptographic core. Jayasinghe *et al.* utilize the Xilin FPGA's Mixed Mode Clock Managers (MMCM) primitive to produce up to 3,072 unique frequencies between 24MHz and 48MHz [43]. All these distinct frequencies are classified into three groups. Each group of frequencies is output through a MMCM primitive's output clock port. The cryptographic core's clock cycle is then driven by one of these three clock signals at random. Jayasinghe *et al.* suggested that by carefully selecting separate frequencies for each clock cycle, their protected standalone AES-128 implementation could achieve 67,684 encryption completion times ranging from 208.33*ns* to 833.32*ns*. The authors reasoned that the greater the number of distinct completion times is, the greater the misalignment existed in the observed traces. As a result, the resistance against power analysis attacks was increased. Jayasinghe *et al.* reported that even with four million processed traces, the secret key are not revealed by conventional CPA attacks. Additionally, the TVLA test using one million-trace did not detect any side channel leakage. This countermeasure is named as the Runtime Frequency Tuning Countermeasure by these authors (RFTC). Jayasinghe *et al.* later apply a similar method called SCRIP to the LowRISC open-source RISC-V processor [44]. Software implementation of AES-128 are executed by the protected SCRIP LowRISC. The protected design is shown to have no first-order leakage in the TVLA test result using two hundred thousand power traces. It also withstood conventional CPA attacks using three hundred thousand power traces. In 2020, a similar prevention technique called Dynamic Frequency Randomization (DFR) is proposed by Hettwer *et al.*[45]. This technique takes advantage of the clock manager IPs on FPGAs' on-the-fly capability to generate about 2,000 unique frequencies. However, their proposed solution is tied to advanced system-on-chip (SoC) architectures that include a Programmable Logic (PL) part and a Processing System (PS) part. The PL part is drove by a dynamic input clock that generated by the PS part. In turn, the PL part produce a highly randomized output clock from its input clock signal. This highly randomized clock is used to drive the target cryptographic circuit, in the same strategy as the proposed techniques from these authors' previous works [43, 44]. Hettwer *et al.* claimed that their protected design was able to obtain over 20 million distinct AES-128 encryption completion times. They performed security evaluation by implementing the protected design on a Xilinx's Zynq UltraScale+ FPGA. The experimental results showed that their protected design is capable of withstanding the profiled DL-SCA attacks, five million-traces TVLA test, and conventional CPA attacks.

In these prior works, after each clock cycle, the clock frequency of the target crypto-core is randomly changed. The corresponded authors consider the most critical designing parameter is the number of unique encryption completion times, which is defined as the total of the numerous clock cycles required by a crypto-core to perform an encryption operation. Their works strive for the greatest variety of distinct completion times possible. Using Encryption completion times as the most critical design parameter, in our opinion, is improper for the several reasons. First, the adversaries only interest in the targeted devices' instantaneous power consumption when the it executing the target sensitive intermediate values. If the PoIs are well-aligned within the set of obtained power traces, the PA attacks have a better chance to succeed. For example, we can consider a hardware implementation of AES-128 that could complete each encryption in ten clock cycles as. The attacker can choose the first round's S-Box substitution output or the last round's S-Box substitution input as the target intermediate values [3]. In both circumstances, the POI is positioned in the very first or in the tenth cycles, respectively. Furthermore, different completion times still could have aligned POIs in obtained power traces. For instance, when the intermediate values is the first round's S-Box substitution outputs, POIs can still be aligned even though the completion timings are different, if the very 1st clock cycles of each encryption has the same frequency and other followed clock cycles have different frequencies. Hence, the suggested RDFS countermeasure in this thesis tries to produce as many different frequencies as feasible while maintaining a constant frequency of the cryptographic circuit throughout each cryptographic process. Only in that case, we can be certain that the POIs all observed power traces are significantly out of synchronization.

5.3 Experiments

5.3.1 Experimental setup

5.3.1.1 Test device

The target device used in subsequent experiments are a 32-bit RISC-V SoC. It is a variation of the secure RISC-V platform described in our earlier study [76]. It has multiple cryptographic hardware accelerators to accelerate the Keystone Trusted Execution Environment's boot operation. The Chipyard framework is used to create the target System-on-Chip [77]. This framework offers designers with a variety of hardware generators and utilities (written in Chisel programming language) to be used in developing customized RISC-V SoCs. Figure 5.1 illustrates the architecture of the target System-on-Chip. The SoC is built around a RV32IMAC Rocket core. The term RV32IMAC means that the target SoC supports several extension of the RISC-V ISA, including the **I**nteger extensions, the **M**ultiplication extensions, the **A**tomic extensions, and the **C**ompress extensions. The target Rocket core is equipped with 16KB of Layer-1 data cache and 16KB of instruction cache to form the RocketTile core complex. A network of interconnect bus is connected to the core complex using the TileLink protocol [78]. This network consists of a Peripheral bus, a Control bus, a Memory bus, and a System bus. In turns, related

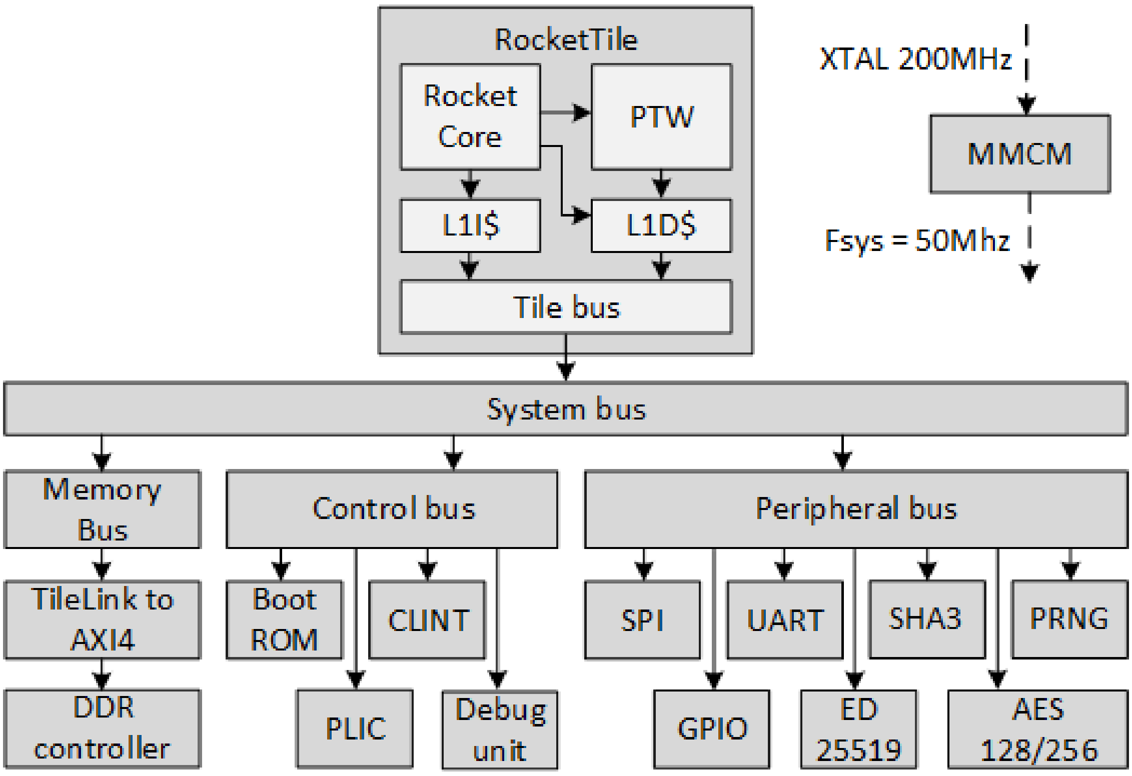


Figure 5.1: System architecture of the experimental RISC-V SoC.

hardware modules are attached to each bus. The Control bus is connected to a variety of peripherals, including the Debug unit, the Platform-level Interrupt Controller (PLIC), the Core-Local Interrupts (CLINT), and the Boot-ROM. The Memory bus connects with the external memory controller, which can support 1GB of DDR3 RAM. This external RAM serves as memory space for running the Linux OS.

The target System-on-Chip employs a three-stage boot mechanism similar to the bootloader of the Freedom U540-C000 platform [79]. The very first boot stage called the zero stage bootloader (ZSBL) is stored in the 16KB of Boot-ROM. The ZSBL are the very first instructions that must be processed when the target SoC comes out of reset state or being powered up. The target SoC can be externally controlled via standard JTAG protocol using the Debug unit to write or read data/instruction to or from system's memory. The PLIC is used to combine and mask all interrupts from external or internal hardware source. Software and timer interrupts are handled by the CLINT. On the other hand, other memory-mapped input/output (MMIO) peripherals are connected to the system via Peripheral bus. These MMIO peripherals include controllers for the Universal Asynchronous Receiver-Transmitter (UART), the General-purpose input/output (GPIO), and the Serial Peripheral Interface (SPI), as well as a number of different hardware accelerators such as the Pseudo-Random Number (PRNG), the Edwards-curve Digital Signature Algorithm (Ed25519), the Secure Hash Algorithm (SHA3), and the Advanced Encryption Standard (AES128/256). The next two stages of boot mechanism, called First Stage Bootloader (FSBL) and the Linux bootloader are stored on an external SD-card. The target SoC communicates with the external SD-card by using the SPI controller. The UART controller enables communication between the target System-on-Chip and other devices using the UART protocol. The secure three-stage boot mechanism involves several cryptographic operations. Therefore, the hardware accelerators are used to speed up the boot process as described in our previous work [76]. They are inherited and used as noise sources in later experiments to demonstrate the AES module's vulnerability even when it is integrated into an unprotected, complicated System-on-Chip. The AES core is an open-source RTL design that can be found on Github [80]. The maximum clock frequency of this standalone AES core is around 100MHz when implemented on FPGA.

As indicated previously, the target System-on-Chip uses a bootloader similar to the one used by the Freedom U540-C000. As a result, it may run in Linux operating system like a microprocessor or operate in bare-metal mode like a simple microcontroller. Users can use the Linux applications or the bare metal programs to control and use the AES core. When the target System-on-Chip is operating in bare-metal mode, only a waiting loop is processed by the Rocket core while the encryption/decryption are performed by the AES core. Therefore, in this case, the unrelated switching noise in power traces is kind of predictable. On the other hand, when the target System-on-Chip is operating in OS mode, additional background processes would be executed concurrently with the encryption/decryption of the AES core. The switching noise is more unpredictable in this scenario. As a result, conducting power-based side channel attacks on the target System-on-Chip while it is in Operating System mode is more challenging than conducting power analysis attacks while it is operating in bare-metal state.

5.3.1.2 Test device with RDFS countermeasure

To begin, the Mixed-mode Clock Manager primitives need to be considered. It is a FPGA primitive from Xilin. The MMCM can be used to generate multiple output clocks with a wide range of frequencies using a input clock signal that has fixed frequency.

Additionally, the MMCM can also be used as a to filter out the jitter in output clock signals. In later experiments, the target System-on-Chip is implemented into Sakura-X FPGA board. This board consists of a Xilinx's Kintex-7 FPGA, which has 8 MMCMs available. A MMCM primitive is a hybrid of analog and digital circuitry. It may be dynamically adjusted by writing appropriate values to its controlling registers to modify the frequency, phase shift, and duty cycle of the output clock. The block diagram of an Kintex-7's MMCM primitive is presented in Figure 5.2. The MMCM primitive can simultaneously produce up to seven clock outputs. A programmable counter divider (D) divides the input reference clock. Counter D is an integer counter that counts between 1 and 106. The Phase-Frequency Detector (PFD) compares the phase and frequency of the input reference clock to that of the feedback clock. The control signal of the Loop Filter (LF) and the Charge Pump (CP) is produced by the PFD. This control signal is proportional to the mismatch in term of phase and frequency between the feedback and input clocks. The CP and LF, in turn, provide a reference voltage that will be used by the Voltage Control Oscillator (VCO). Next, the VCO provides a clock signal with high-frequency and distributes it to 8 programmable output counters ($O_0, O_1, ..., O_6$, and M). The counters $O_0, O_1, ..., O_6$ function as frequency multipliers and generate seven output clocks, while the counter M generates a feedback clock signal. The counters $O_0, O_1, ..., O_6$ are integer counters that can count integers from 1 to 128, whereas the counters O_0 and M are fractional counters with 0.125 increments that can count from 2 to 128 and from 2 to 64, respectively. Thus, the output clock associated with O_0 has finer resolution than those associated with other $O_1, O_2, ..., O_6$ outputs. Fine-phase shifting is also supported since the VCO provides 8 fixed phase variants and 1 varying-phase variant of the output clock signal. The Mixed Mode Clock Manager also designed with other controlling registers to adjust the phase and duty cycle of each final output clock signal. However, we are just concerned with the output clock signals' frequency in this work.

While the programmable counters built within the Mixed Mode Clock Manager primitive offer a large range of counting values, several constraints must be adhered to. These constraints are recommended by Xilinx to ensure the MMCM's stability, and they are detailed in the corresponding FPGA chip's Data Sheet. The experimental platform is a Kintex-7 FPGA in this study. Hence, these important constraints can be found in this

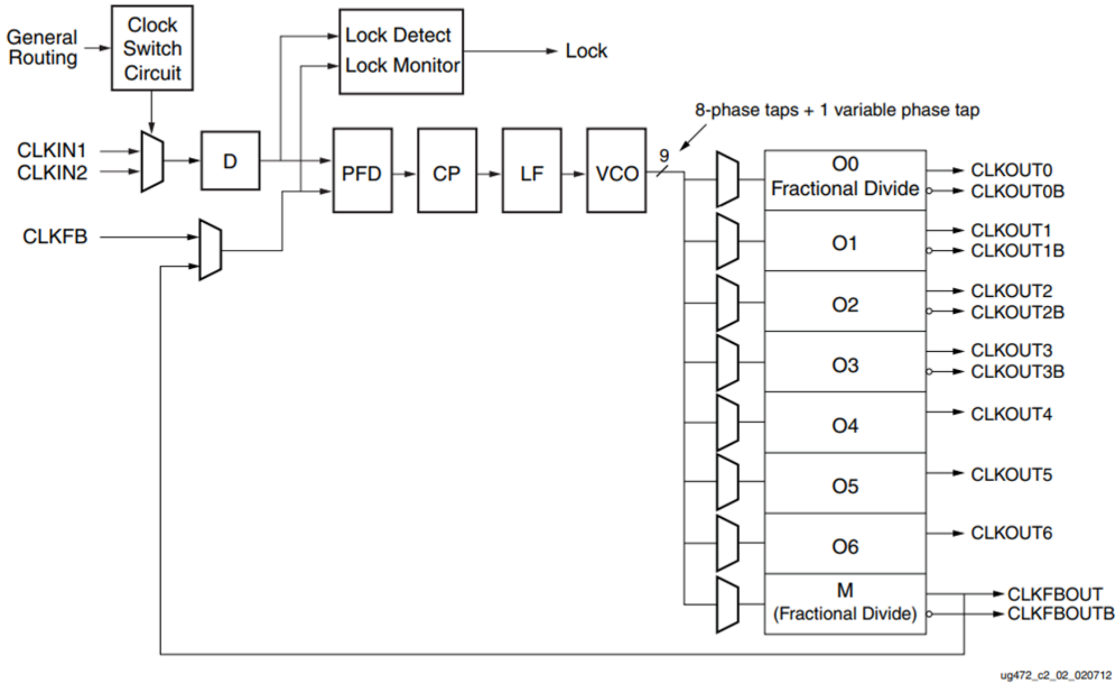


Figure 5.2: Xilinx 7-Series' Mixed Mode Clock Manager [81].

reference [82]. The following constraints must be met, according to [82].

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

The output clock frequency of the Mixed Mode Clock Manager is calculated by using (5.1).

$$F_{CLKOUT_i} = \frac{F_{in} \times M}{D \times O_i}; i \in (0,6) \tag{5.1}$$

To strengthen the targeted RISC-V System-on-Chip's resistance to power analysis attacks while keeping minimal overheads and limiting the overall System-on-Chip's performance degradation, the RDFS countermeasure is exclusively proposed for the target cryptographic core, which in this case is the MMIO AES peripheral. The scaled clock frequencies are dynamically produced by using the Mixed Mode Clock Manager. Multiple output ports a single Mixed Mode Clock Manager primitive have been employed concurrently in prior researches [43, 44, 45]. These outputs are generated exclusively when the integer counter mode is used. We propose to use the fractional output clock

O_0 to significantly improve the number of possible scaled frequencies in our proposed method. Besides, in our proposal, the clock frequency is not changed after each clock cycle. Rather than that, the cryptographic accelerator's clock signal is updated only after each encryption/decryption. We can ensure that the misalignment of the POIs in each measured power trace corresponds to a distinct clock frequency by doing so. The number of possible locations for the PoI would exactly be equal to the number of distinct clock frequencies that can be created. Finally, we use the SoC's DDR memory to store all distinct configurations of the MMCM primitive dynamically. In the previous related works, these parameters are synthesized and stored using the Block RAM that available on FPGA. As a result of the FPGA's hardware resource constraints, only a limited number of reconfiguration parameters can be stored.

To install the proposed countermeasure, some modifications to the targeted RISC-V System-on-Chip are required. The modified RISC-V System-on-Chip's system architecture is shown in Figure 5.3. To begin, the RISC-V System-on-Chip is enhanced with the addition of several components, including a pulse counter, an additional MMCM primitive, and a peripheral named Dynamic Reconfiguration Port (DRP). Apart from the 50MHz clock signal F_{sys}, which was used to drive the entire original System-on-Chip, the first original MMCM also produces an 800MHz clock signal F_{in}. Additionally, the TileLink connection between the Peripheral bus and the AES-core is modified into an asynchronous crossing configuration, which enables the AES-core to run in a different clock domain with system's other modules. The Dynamic Reconfiguration Port peripheral is connected to the Peripheral bus using the TileLink protocol as an MMIO peripheral. It consists of a finite state machine and a number of addressed registers. The finite state machine is described in detail in Xilinx's Application Note [83]. The addressed registers contain values for the fractional and integer parts of the O_0, M, and D counters, respectively. The finite state machine calculates the reconfiguration settings based on the addressed registers' values and applies them to the second MMCM. Based on the input clock F_{in} and the received configuration, the second MMCM produces a new clock frequency. The AES accelerator is driven by the produced clock. The created clock is also fed into the pulse counter. The pulse counter will count the rising edges of the generated clock over a predetermined time interval set by the external reference clock of 200MHz. After the predetermined interval, the counted result is stored in an addressed register.

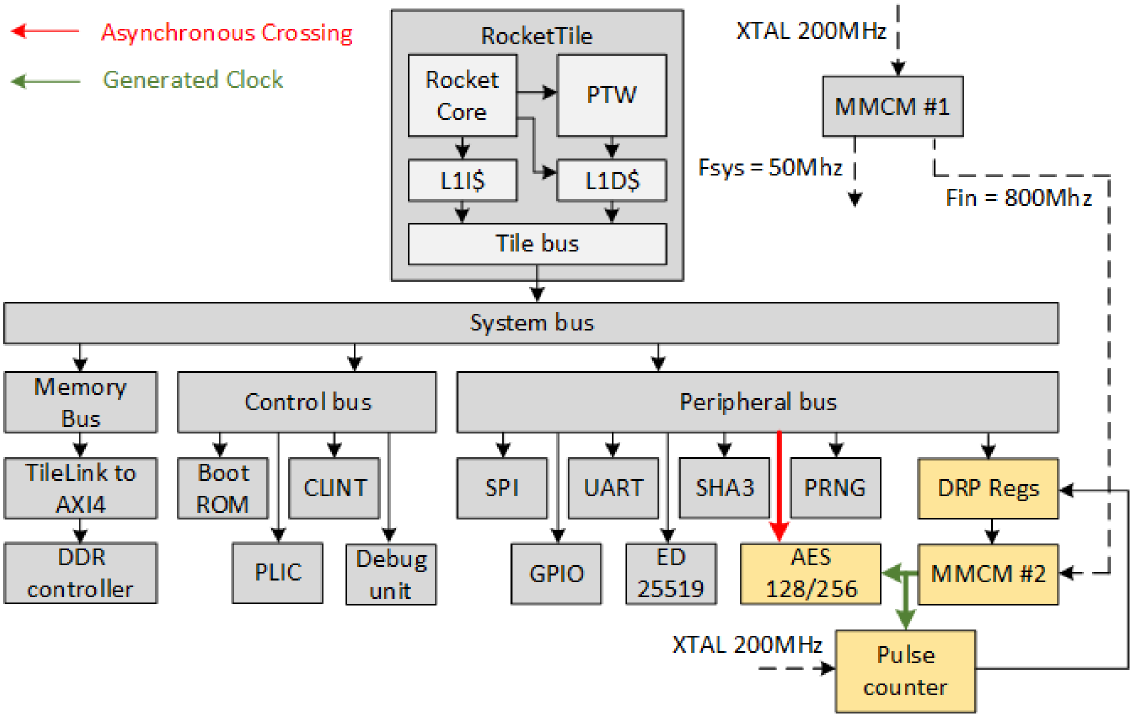


Figure 5.3: System architecture of the modified experimental RISC-V SoC.

The Rocket core can write the suitable O_0, M, and D values to dynamically scale the operational clock frequency of the AES accelerator without resetting the entire System-on-Chip. The pulse counter's results can also be read by the Rocket core to check the generated clock frequency's accuracy.

With all these modification, the updated System-on-Chip presented in Figure 5.3 is capable of dynamically adjusting the operating clock frequency of the AES-core. To properly leverage this ability to resist power analysis attacks, we must produce as many different clock frequencies as possible and use these produced unique clock frequencies to drive the AES-core for each encryption/decryption.

Due to the fact that the MMCM primitive may be adjusted during operation and the output clock frequency can be determined using (5.1), a MATLAB script would be used to study the influence of F_{in} on the number of different output frequencies that can be produced. The MATLAB script considers all constraints associated with $O_0, M, D, F_{VCO}, F_{PFD}, F_{in}$ [82, 81]. Additionally, the AES-core's maximum operating frequency as stated in [80] is used as the upper limit for the MMCM output frequency. Moreover, we wish to keep the proposed countermeasure's time overhead to a minimum. Hence, the MATLAB script also uses the operating frequency of the whole system, 50MHz, as the lower limit. By verify the generated clock frequency's accuracy, we found that the precision of the generated clock is ±1 Hz. The relation between the number of different output clock frequencies that can be produced and the input clock frequency of the MMCM is illustrated in Figure 5.4. The Mixed Mode Clock Manager can produce 219,412 different clock frequencies when being fed an input frequency of 800MHz. Hence, the first original MMCM is configured to generate an additional 800MHz output clock signal, This 800MHz clock will be used as the F_{in} of the second MMCM. The MATLAB script also generates all 219,412 possible combinations of O_0, M, and D counters, each of which corresponds to a different output clock frequency, and stored them in a C header file. Five bytes are required to represent each possible combination. The O_0 must be represented using two bytes. The fractional M counter is represented by two bytes, one for the fractional and another for the integer part. One byte represents the integer D counter. Therefore, approximately 1.05MB is required to represent all 219,412 generated combinations.

Lastly, a software is created to control the operation of the proposed RDFS countermeasure. It would declare the C header to include and use all 1.05MB of pre-computed MMCM combinations. All MMCM combinations and the control software and are compiled and stored in the external DDR RAM during runtime. Figure 5.5 illustrates the program's flowchart. The PRNG generates a random number prior to each AES encryption/decryption. The generated random number is compared to 219,412, which is the total number of MMCM combinations. The random number is discarded if it exceeds 219,412 and a new random number is generated by the PRNG. This process is repeated until a number that smaller than 219,412 is generated. The control program utilizes this number as an index to find the corresponding MMCM configuration. All related O_0, M, and D counter's values are written to the DRP peripheral so that a new clock frequency is generated and applied to the AES-core. When the Lock signal is asserted by the MMCM, that means the generated clock is steady, the AES-core resumes its normal encryption/decryption operations.

5.3.1.3 Automatic power trace measuring system

An experimental system is configured to automatically acquire power traces. The experimental system is shown in Figure 5.6. It is composed of a Monitoring PC, a Sakura-X FPGA board serving as the test platform, and a Tektronix MSO2024B oscilloscope.

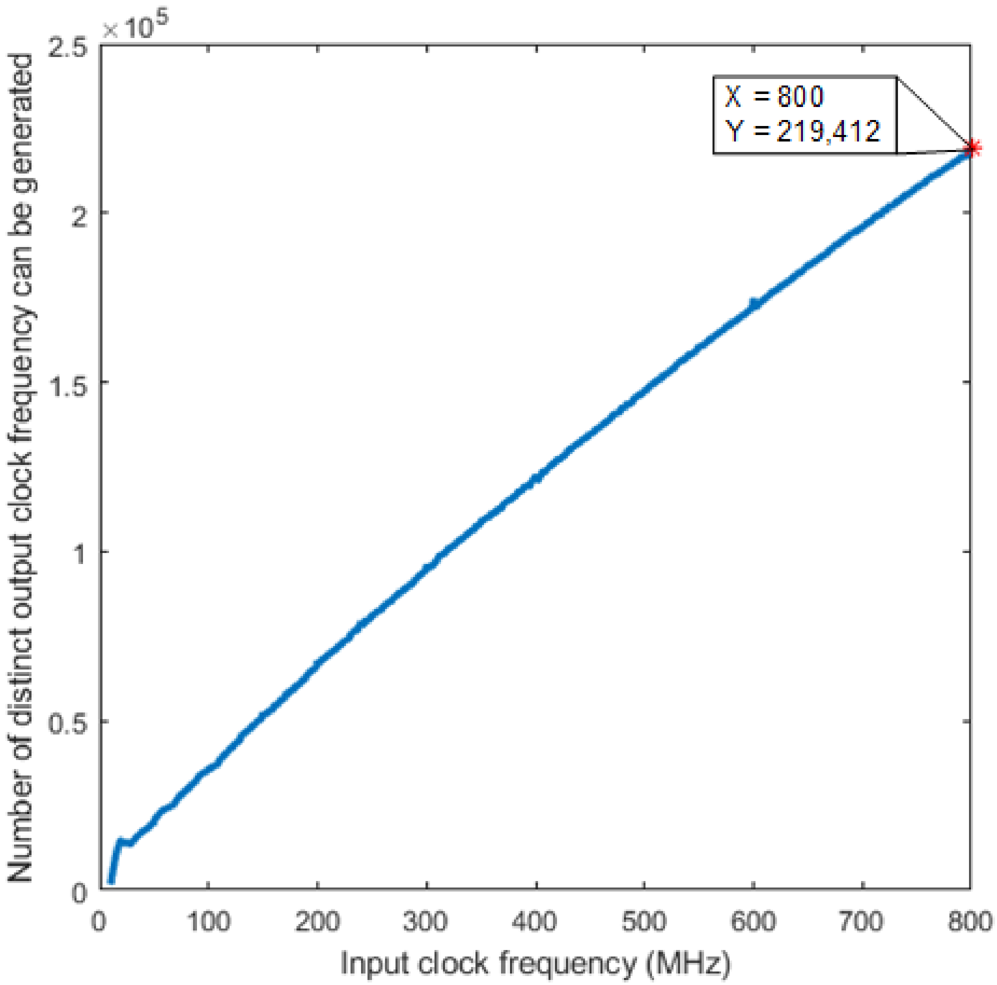


Figure 5.4: The relationship between frequency of input clock and the number of possible distinct output clock frequencies.

There are two separate FPGA chips on the Sakura-X board, including a Xilinx's Spartan-6 XC6SLX45 and a Xilinx's Kintex-7 XC7K160T. Additionally, there are probe points and a shunt resistor on the Kintex-7 FPGA's core VDD line. Therefore, we implement the entire targeted RISC-V System-on-Chip on the Kintex-7 FPGA and monitor the variation of the logic core's VDD. These variation are recorded and saved as power traces Additionally, a signal from the AES accelerator's status register is extracted and mapped to an FPGA pin. It will produce an square pulse when the AES-core starts its encryption/decryption. This pin will be used as trigger signal for the oscilloscope's acquisition process.

The Tektronix MSO2024B oscilloscope is used to capture power traces generated by the targeted RISC-V SoC when AES-128 encryptions are processed. Four analog

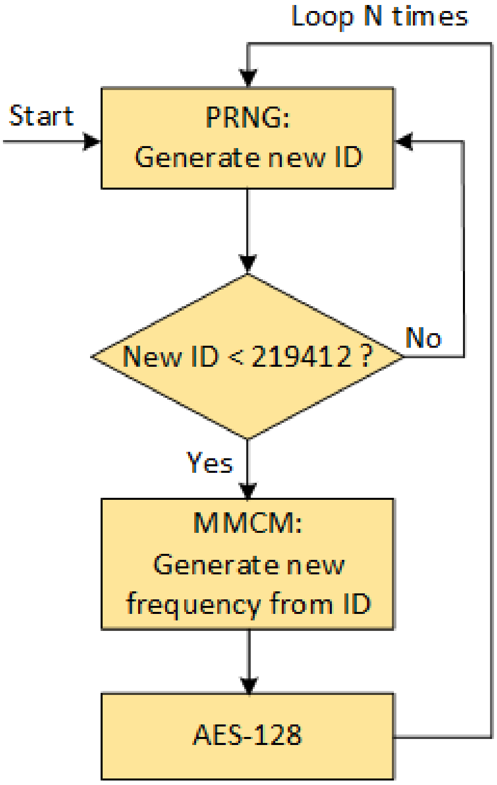


Figure 5.5: Flowchart of control program.

channels are included in this oscilloscope. The highest sampling rate is 1 GS/s and the maximum bandwidth is 200MHz. The measurement is made using two passive probes. One probe is utilized to detect the target SoC's trigger signal. The other probe acquires the analog signal from the Kintex-7 FPGA's core V_{DD} node. Through VISA Virtual Instrument Software Architecture (VISA) communication, the Monitoring PC can remotely control this oscilloscope.

The Monitoring PC is responsible for the overall operation of the auto-measuring system. The PC communicates with the oscilloscope via USB and with the target SoC via UART. It repeatedly delivers plaintexts to the target System-on-Chip and instructs the oscilloscope to capture power traces for each encryption performed by the target System-on-Chip. After each encryption, the Monitoring PC receives the oscilloscope's measured power trace and the associated ciphertext from the target System-on-Chip. Additionally, the Monitoring PC verifies the ciphertext to guarantee that the target System-on-Chip correctly encrypts the plaintext. The power trace, as well as the plaintext and ciphertext associated with it, are recorded for later analysis.

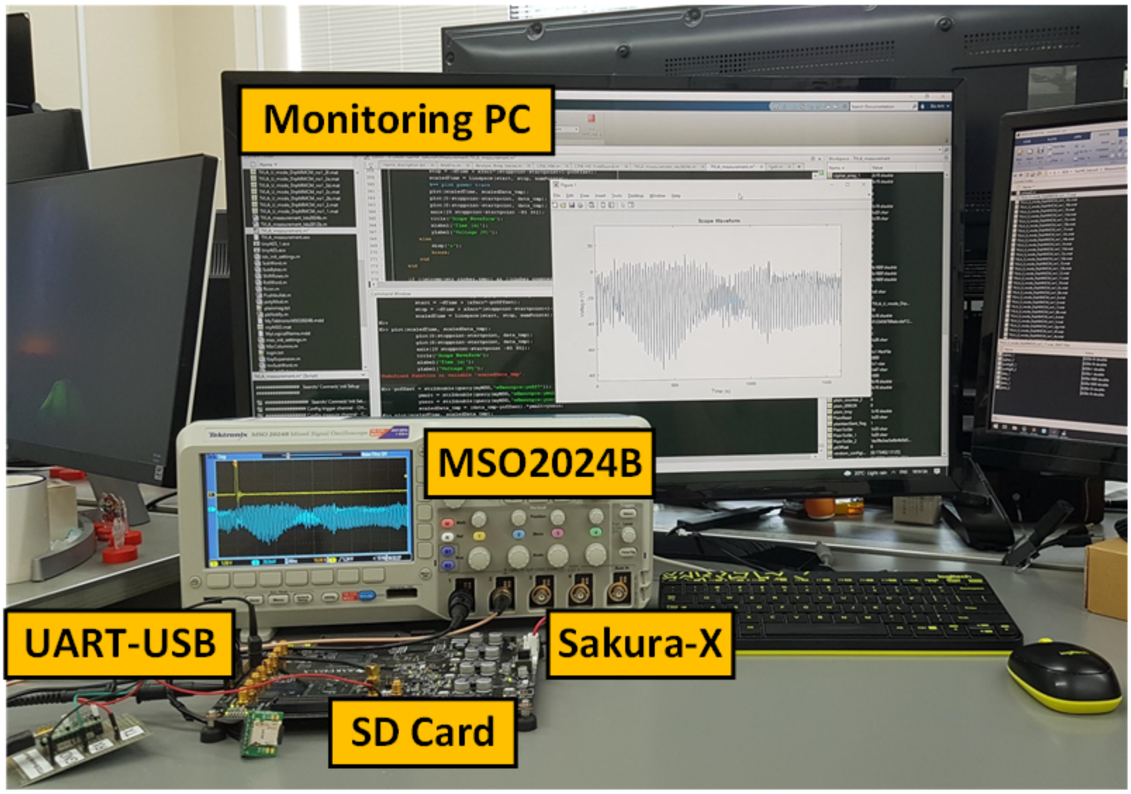


Figure 5.6: Experimental system.

5.3.1.4 Power trace analysis

5.3.2 *Experimental results*

5.3.2.1 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 implemented SoC can operate at a maximum clock frequency of 50MHz. The 50MHz clock signal for the SoC system is generated by an MMCM primitive using an external 200MHz crystal oscillator. Following that, the modified RISC-V SoC architecture described in Section 5.3.1.2 is implemented into the Kintex-7 XC7K160T FPGA, thereby replacing the unprotected design. The pulse counter, which was used to verify the accuracy of the generated clock during the design phase, is now unnecessary and has been eliminated. The following Table 5.1 summarizes the post-implementation utilization of the two RISC-V SoCs. In a Kintex-7 XC7K160T FPGA device, the integrated AES128/256 accelerator consumed just 3.13 percent of the total available Look-up table (LUT) and 1.63 percent of the total available Flip-flop (FF). Meanwhile, the entire unprotected SoC consumes 48.31 percent of the LUT and 19.38 percent of the FF, respectively. In other words, the AES accelerator accounts for less than 8.5 percent of the total hardware utilization of the SoC. Moreover, the table demonstrates that with the addition of hardware for implementing the RDFS countermeasure, the number of utilized LUT and FF increases by just 4.2 percent and 0.55 percent, respectively. An additional MMCM primitive is also used in the protected design.

Table 5.1: Xilinx's Kintex-7 utilization results at post-implementation stage.

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

Additionally, we measure the execution times of tens millions AES-128 encryptions from protected and unprotected designs. These encryptions are performed by the corresponded AES accelerators. The AES accelerator in the unprotected design operates at 50MHz, whereas the protected design operates at a randomly chosen 219,412 clock frequencies in the range of 50MHz to 100MHz. The results of the measurements indicate that the average time overhead is 3.36 times. Although the AES accelerator was operated at higher rates, timing penalties are still existed due to the generation of random indices and associated clock frequencies. We use a basic Linear Feedback Shift Register (LFSR)-based PRNG in this study. This PRNG is embedded as a peripheral and is used to produce random values which are used by other cryptographic peripherals (SHA3, Ed25519). The time overhead would be reduced if a dedicated PRNG was included specifically for the RDFS countermeasure, as other comparable efforts have demonstrated [43, 44, 45]. Without accounting for the time penalty associated with producing the random indices, the time overhead is only 1.83 times.

5.3.2.2 Test Vector Leakage Assessment

To begin, TVLA testing is conducted on the unprotected RISC-V SoC. The TVLA tests require the acquisition of power traces as stated in subsection 3.4. The two TVLA test results with two separated groups of power traces are given in Figure 5.7. Each group contains five thousand power traces for DataSet-1 and five thousand power traces for DataSet-2. In both tests, nearly all t-score values surpass the ±4.5 limit, and the absolute values exceed 90, indicating that the unprotected SoC is especially vulnerable to power analysis attacks. With the addition of additional power traces to the TVLA test, the absolute peaks become much higher. The results of the test with fifty thousand power traces are shown in Figure 5.8. The maximum t-absolute score's value increases to 180.

After that, TVLA tests are conducted on the RISC-V SoC that is protected by our proposed RDFS countermeasure. However, as a preliminary step, we gather power traces for the AES accelerator when only 1024 different frequencies are applied. The results of the TVLA test in this scenario are presented in Figure 5.9. A dataset of ten thousand power traces is used in this TVLA test. Although there are still a few points in the t-score trace with absolute values more than 4.5, the maximum t-score has decreased dramatically to about 14. As expected, the RDFS countermeasure with just 1024 unique frequencies has a modest effect on side-channel leakage reduction.

Finally, we apply the RDFS countermeasure with all 219,412 possible distinct clock frequencies. There are no points on the t-score trace that surpass the ±4.5 limit during the middle third of the AES operation in either of the TVLA tests, each on five million power traces, as demonstrated in Figure5.10. In other words, the SoC protected with 219,412 distinct clock frequencies passed the leakage test. Thus, we conclude that there is no evidence of information leakage in five million power traces.

5.3.2.3 Correlation Power Analysis attacks

To assess the effectiveness of the proposed RDFS countermeasure, realistic CPA attacks are conducted. The experimental system described in subsection 5.3.1.3 is used to obtain power traces. In this research, we conduct numerous CPA attacks on the same set of measured power traces, targeting all 16 bytes of the AES-128 accelerator's secret encryption key. The Partial Guess Entropy (PGE) is used to assess the results of the CPA

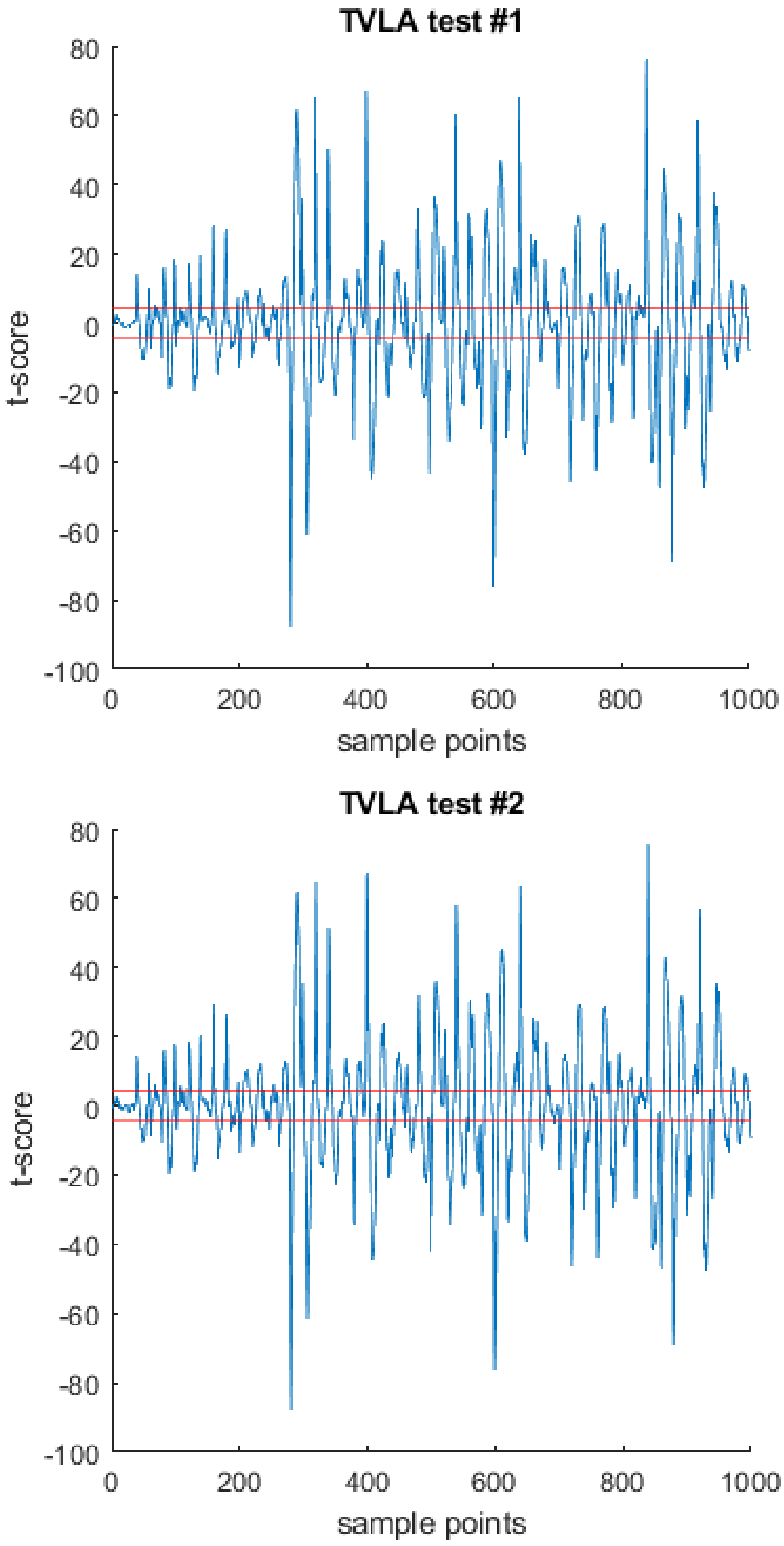


Figure 5.7: Result of TVLA test on unprotected SoC with ten thousand traces.

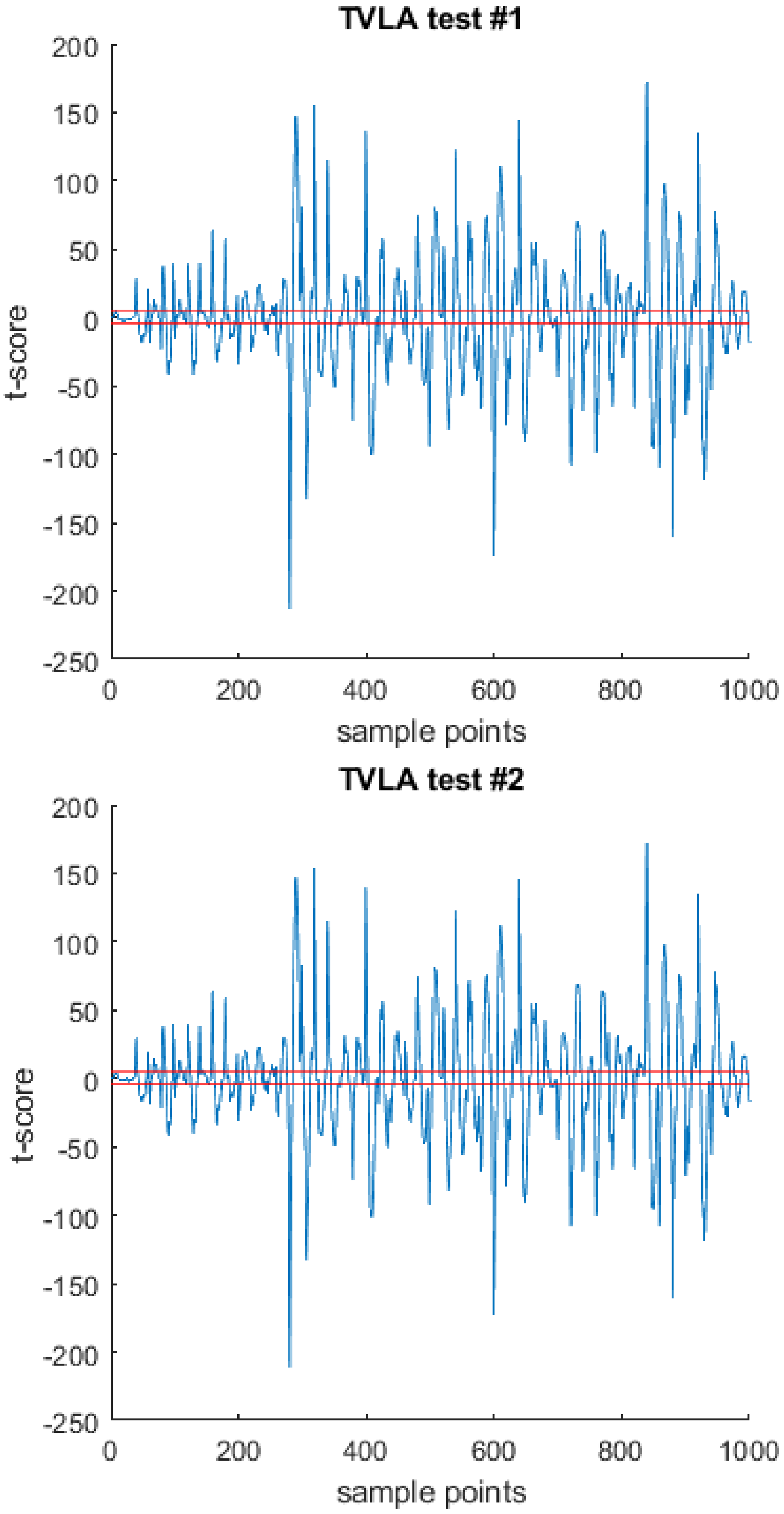


Figure 5.8: Result of TVLA test on unprotected SoC with fifty thousand traces.

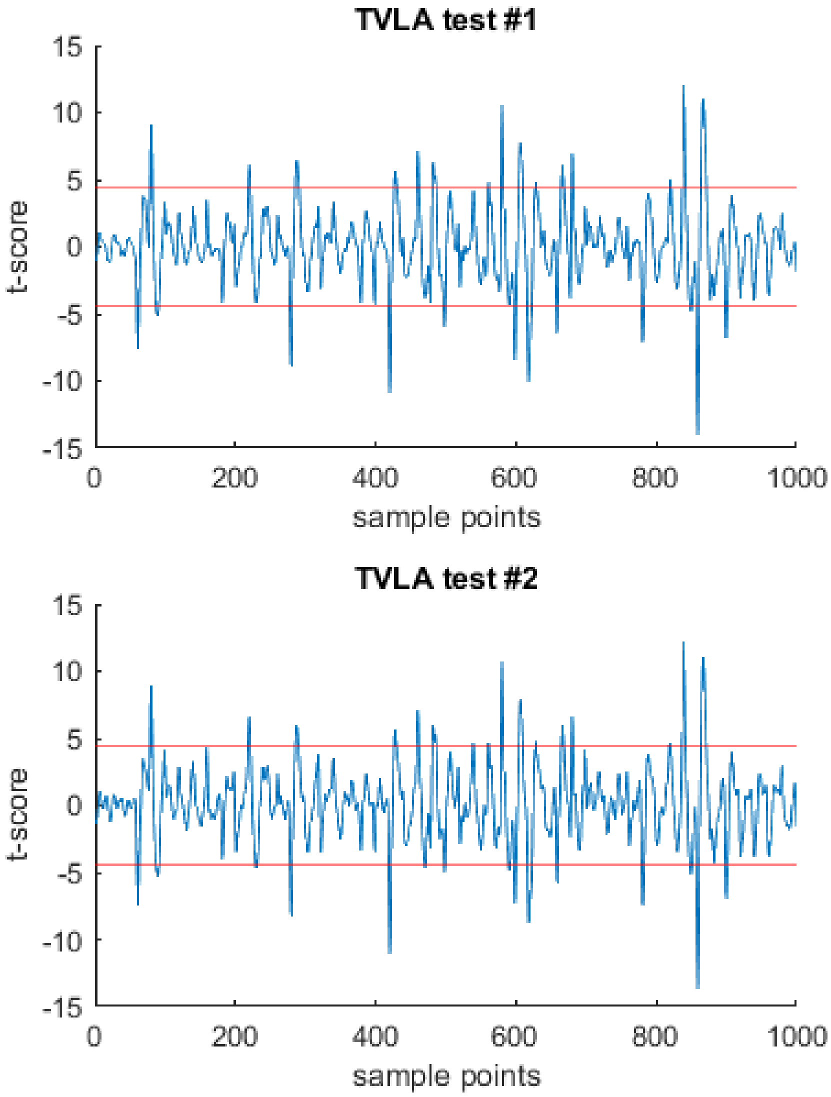


Figure 5.9: Result of TVLA test on RDFS-1024 protected SoC with ten thousand traces.

attack [67]. It is the correct subkey's top-down ranking when all key hypotheses' statistical comparison results are arranged 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.

The results of CPA attacks on the unprotected SoC operating in bare-metal mode are shown in Figure 5.11 These attacks employ seventy thousand power traces. The PGE results eventually reach "0" when attacking 12/16 subkeys, which means the CPA attacks failed to reveal four subkeys (byte numbers 14, 10, 6, and 2) while success with the other twelve subkeys. The minimum number of necessary traces to reveal the correct

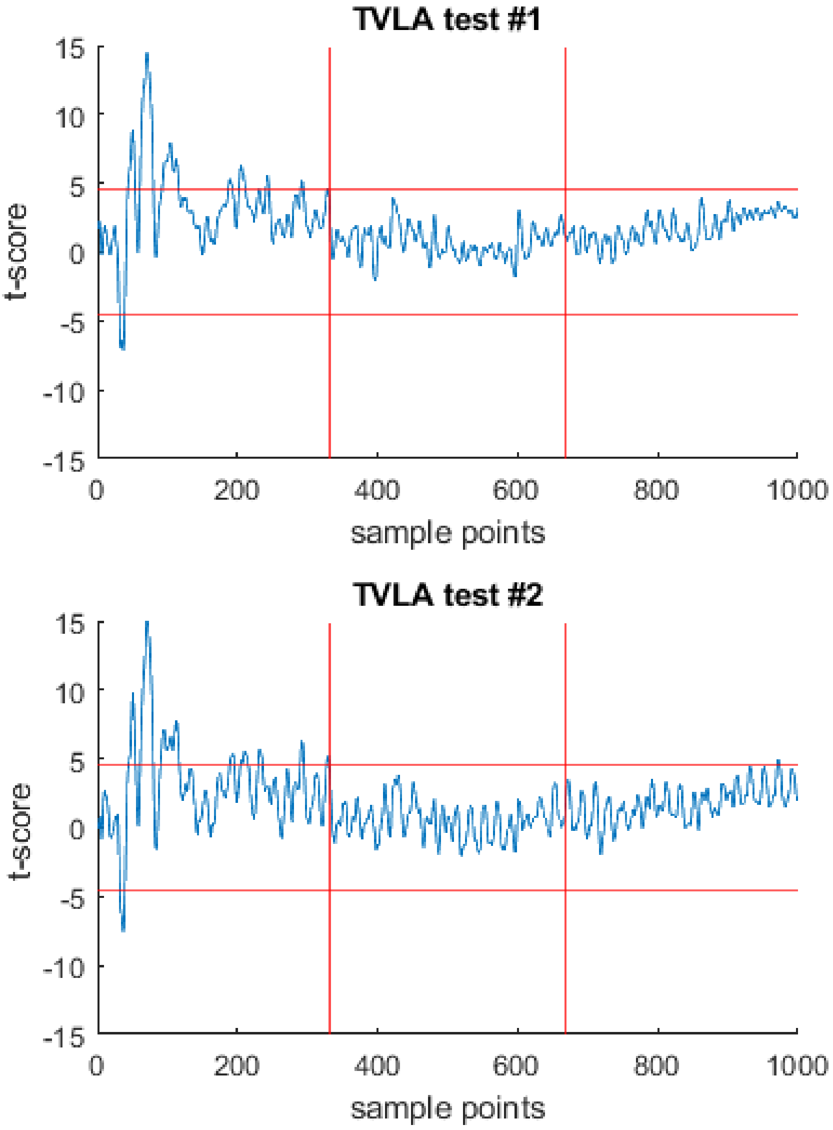


Figure 5.10: Result of TVLA test on RDFS-219412 protected SoC with five million traces.

subkey varies between 1,642 and 58,685 traces, with an average of 28,636 traces. We also attempted CPA attacks on the failed bytes using up to one hundred thousand traces. However, the PGE of these subkeys never reached “0”.

Because the AES accelerator utilizes less than 8.5 percent of the total hardware on the SoC. These poor CPA attacks results could due to the fact that the measured power traces are too noisy. Therefore, we decided to conduct other CPA attacks with a different set of power traces. This new set is measured using the MSO2024B Oscilloscope’s averaging acquisition mode. Modifications are made to the experimental system described

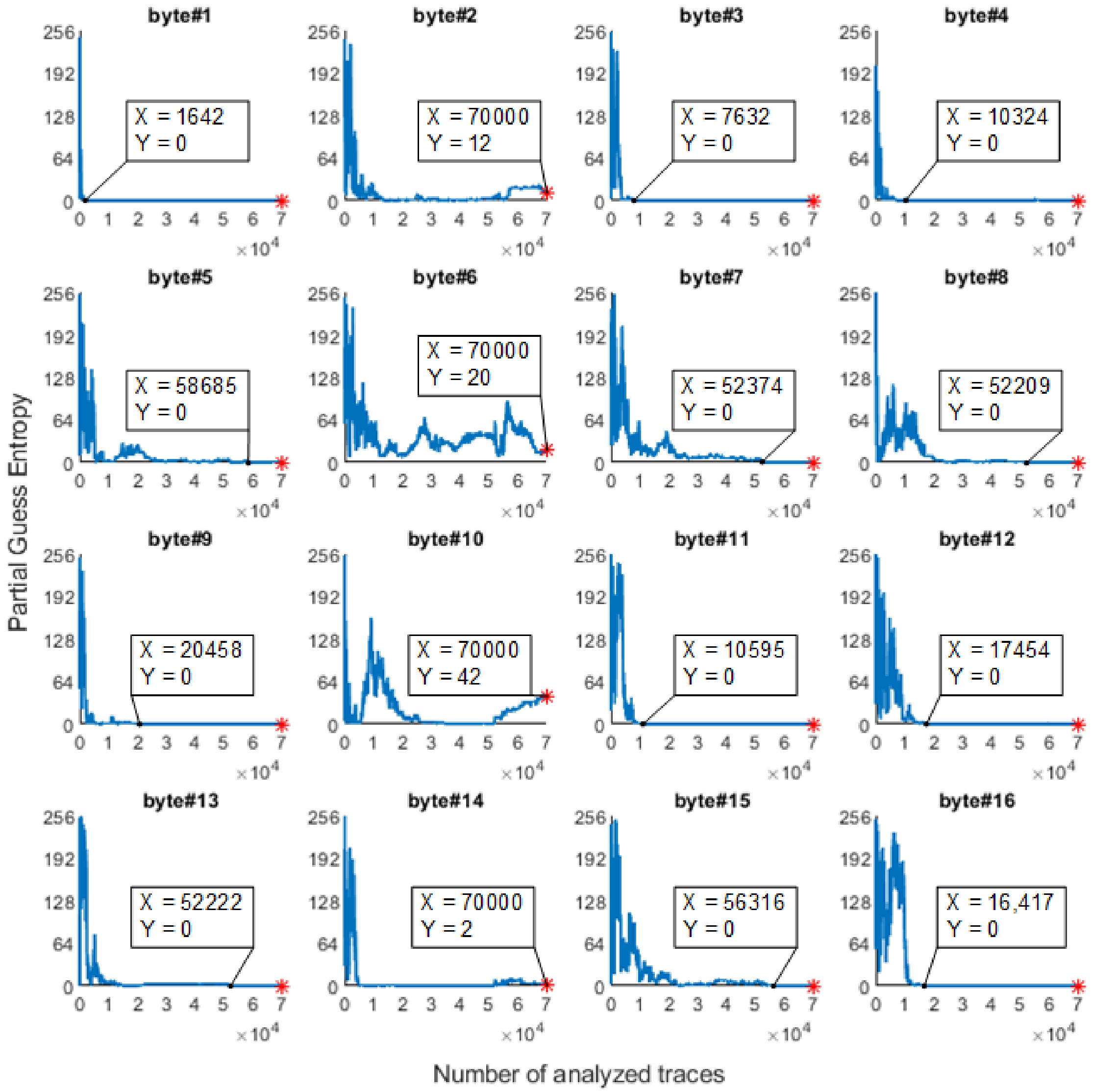


Figure 5.11: Result of CPA attacks on unprotected SoC (bare-metal mode) with seventy thousand traces.

in subsection 5.3.1.3. The monitoring PC sends a plaintext input to the target System-on-Chip 64 times in a row. When the target System-on-Chip performs 64 encryptions on the same plaintext inputs, the Oscilloscope measures 64 corresponding power traces. The average of these 64 traces is generated and used in subsequent CPA attacks. Average measurements could help mitigate the effects of random switching noise generated by components other than the AES accelerator in the SoC. Figure 5.12 illustrates the results of CPA attacks on the unprotected SoC using averaged power traces. Indeed, averaging the traces improves the performance of CPA attacks. Thirteen subkeys are recovered successfully. The minimum number of required traces to recover the correct subkey varies between 465 and 7,613, with an average of 1,928. These CPA attacks, however, are un-

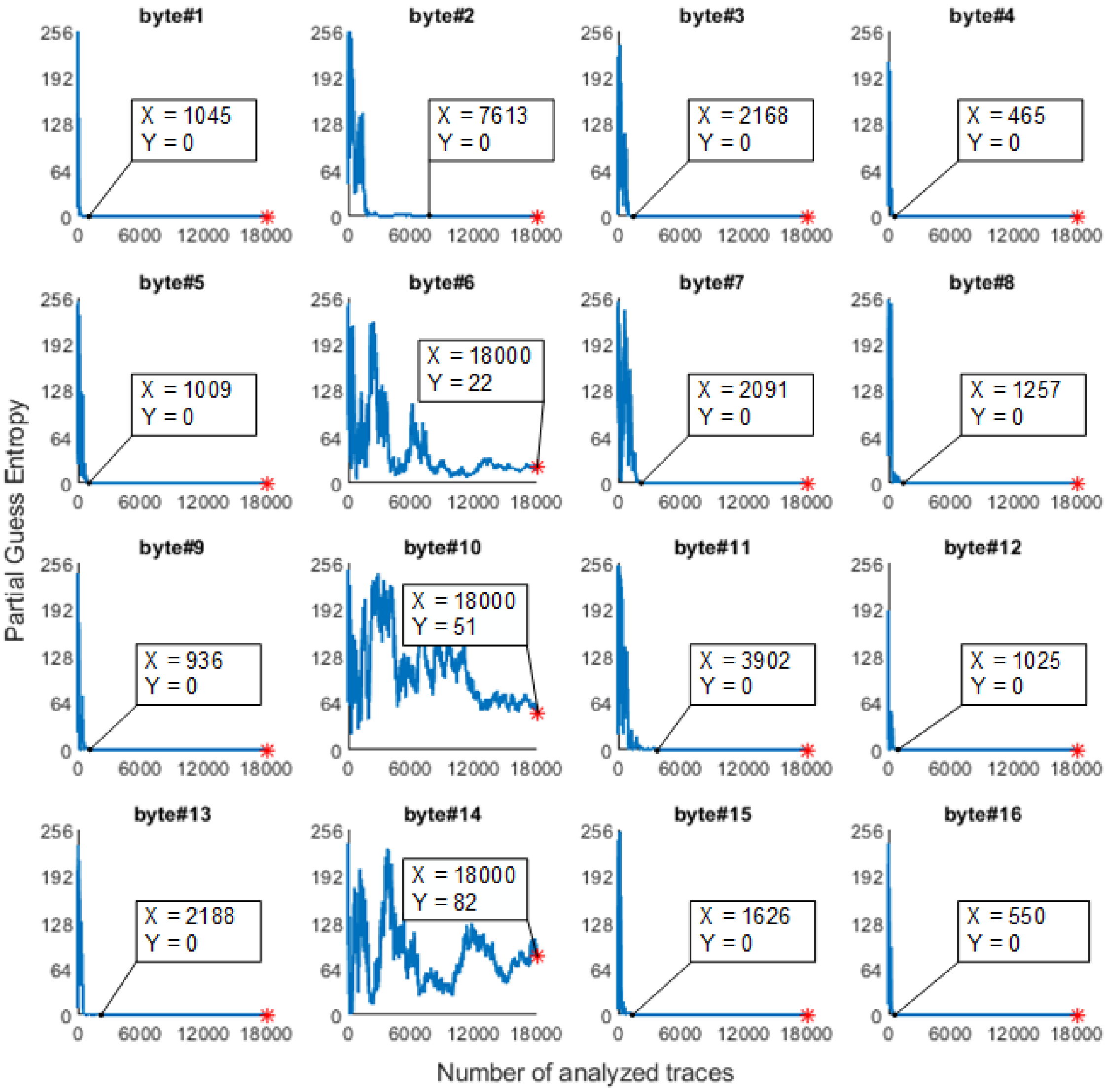


Figure 5.12: Result of CPA attacks on unprotected SoC (bare-metal mode) with eighteen thousand traces measured with average-64 technique.

able to retrieve the correct values for three subkeys (byte numbers 14, 10, and 6). In practice, attackers may conduct brute force attacks on the remaining three subkeys, eventually compromising the security of the entire system.

Additionally, CPA attacks are conducted on unprotected SoCs operating in OS mode. The analysis process employs twenty thousand averaged traces. Figure 5.13 illustrates the related results. These CPA attacks also successfully reveal thirteen subkeys. Similar to attacking the unprotected, bare-metal mode SoC, the remaining uncovered subkeys are byte numbers 6, 10, and 14. However, the minimum number of required traces for disclosing other subkeys is increased in this scenario due to the increased noise in the power traces. The minimum number of required traces varies between 1,650 and 19,591

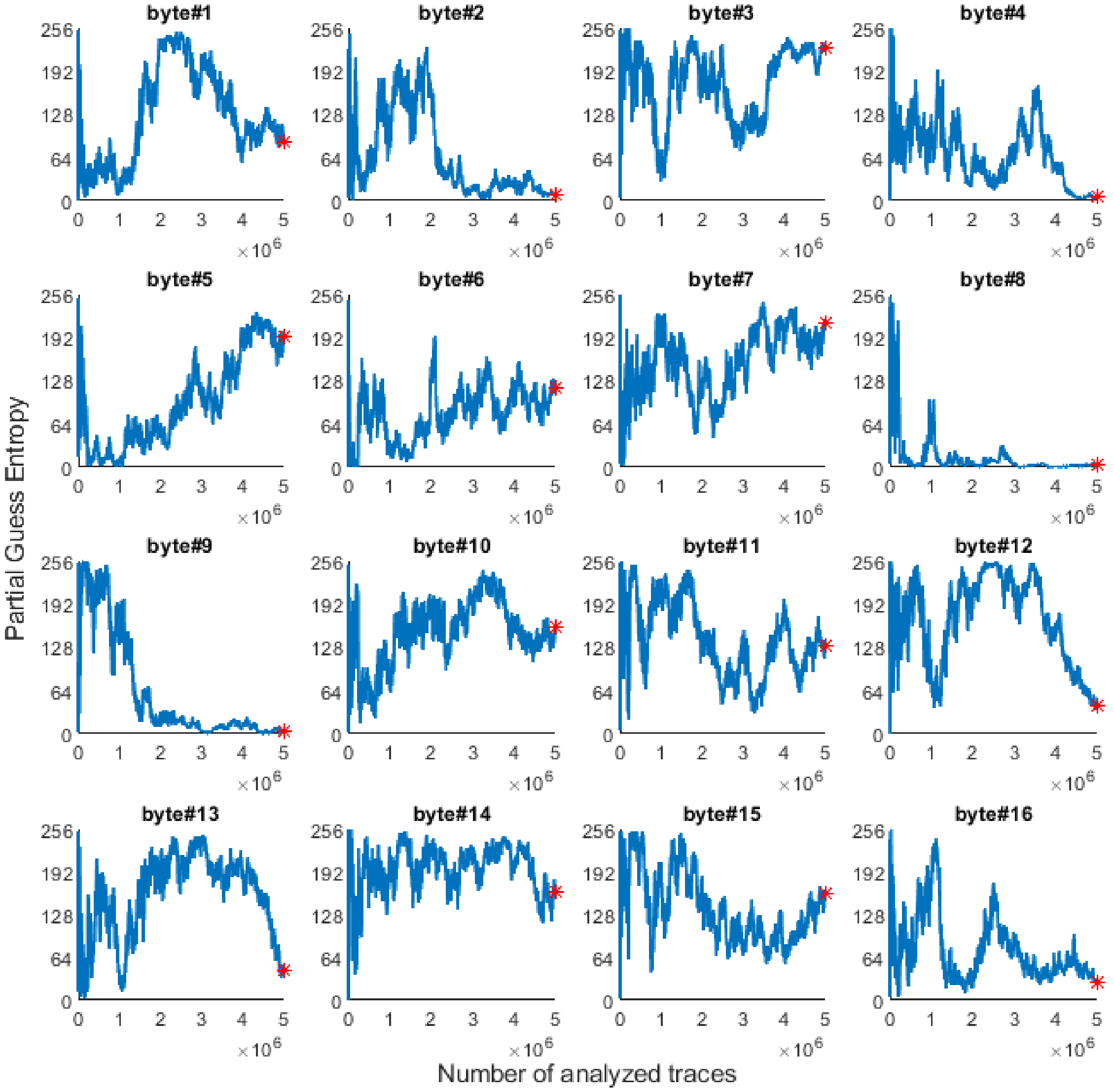


Figure 5.13: Result of CPA attacks on unprotected SoC (OS mode) with twenty thousand traces measured with average-64 technique.

traces, with an average of 10,175 traces.

The results of the CPA attacks on unprotected SoCs indicated that a cryptographic accelerator embedded within a complex SoC is vulnerable to power analysis attacks, even though its size is significantly smaller than the overall SoC's size. Additionally, it demonstrated that the TVLA was capable of detecting side-channel leakage prior to launching realistic attacks.

Following that, we perform CPA attacks against the SoC protected by the proposed RDFS countermeasure, in which the operating frequency of the AES accelerator is randomly changed between 219,412 unique frequencies in the range of 50MHz to 100MHz.

The target System-on-Chip continues to operate in bare-metal operating mode. Figure 5.14 illustrates the results of the CPA attacks. It demonstrates that, despite analyzing five million traces of the protected SoC, none of the PGE results targeting 16 subkeys reach "0". In other words, no secret subkey was recovered during the CPA attacks. Moreover, the proposed RDFS countermeasure prevents CPA attacks on the protected SoC from being conducted using the average measuring technique, as the clock frequency of the AES accelerator is adjusted after each encryption. Therefore, we can conclude that employing the RDFS countermeasure with 219,412 different frequencies considerably improves the target System-on-Chip's CPA resistance. Furthermore, because the TVLA test is unable to detect any leakage with a dataset of five million power traces, we believe that the suggested architecture may withstand CPA attacks with a larger number of power traces. In comparison to attacking unprotected SoCs in bare-metal mode and with average measurement, the number of traces required to successfully attack protected SoCs increased from 1,928 to over five million, a nearly 2,593-fold increase.

5.3.2.4 Profiled Deep Learning based Side Channel Attacks

Recently, it was demonstrated that the state-of-the-art DL-SCA attacks outperformed traditional CPA attacks when it came to exploiting misaligned power traces. Therefore, we also assess our proposed protected design's effectiveness against DL-SCA attacks. Various profiled DL-SCA attacks utilizing the CNN_{best} network architecture [17] are performed on the previously measured power traces.

Table 5.2 summarizes the attack parameters and results. Each DL-SCA attack starts with training the CNN_best model with a fixed batch size and epoch of 200 and 75, respectively. The results confirmed that profile DL-SCA attacks are more powerful than conventional CPA attacks. In attacks on unprotected SoCs operating in bare-metal mode, the DL-SCA can completely derive all 16-bytes secret encryption key. The average number of traces required from the target device is smaller than that required for CPA attacks. When used against unprotected SoCs running Linux OS, DL-SCA attacks perform worse than CPA attacks. There are only nine subkeys that can be exposed. This result corroborates Alipour *et al.*'s assertion that a noise-generation-based hiding countermeasure may provide superior protection against DLSCAs [36]. However, the results indicate that the protected SoC implementing the RDFS countermeasure with 219,412 different clock

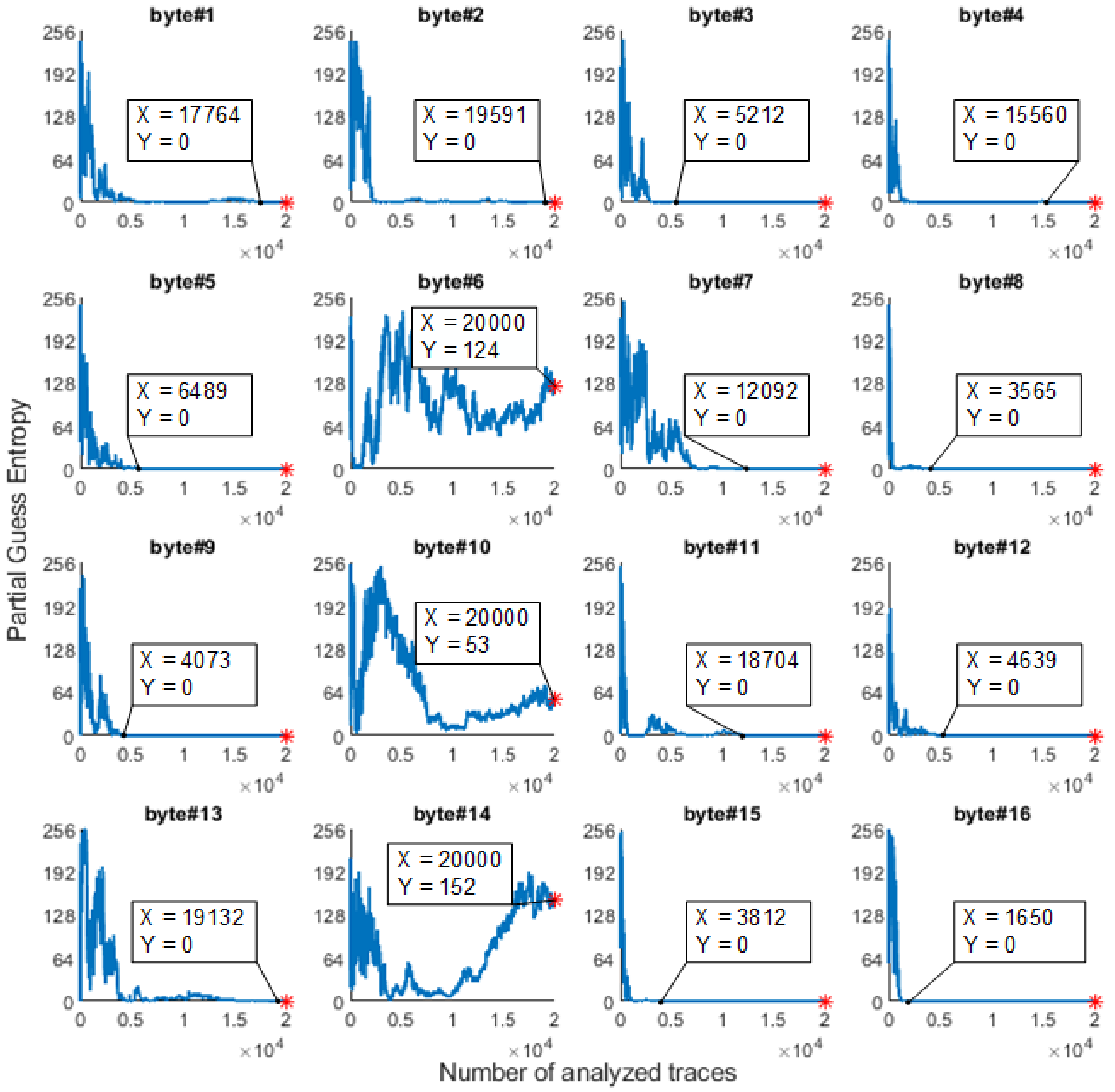


Figure 5.14: Result of CPA attacks on RDFS-219412 protected SoC (bare-metal mode) with five million traces.

frequencies is still vulnerable to DL-SCA attacks. One million traces obtained from the protected SoC are used for profiling, while the remaining one hundred thousand traces are used for attacking. Thirteen subkeys are successfully recovered, with the average number of required traces being 45,924 traces. This finding shows the effectiveness of DL-SCA against misalignment-based hiding countermeasures.

Lastly, in the final DL-SCA attacks experiment, the targeted device is the protected SoC running in bare-metal mode. sixty thousand traces are used in profiling phase, and twelve thousand traces are used in attacking phase, which is similar with the first DL-SCA attacks experiment attacking on the unprotected, bare-metal SoC. When the same number of traces are used, DL-SCA attacks succeed to expose entire secret key used by the un-

Table 5.2: Parameters and results of several profiled Deep Learning based Power Analysis attacks.

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

protected SoC, and failed to recover any subkeys used by the protected SoC. Therefore, we can conclude that while the proposed RDFS countermeasure does not totally prevent DL-SCA attacks, it does contribute to the protected device's resistance.

5.3.3 Results comparison

Our proposed technique's effectiveness is compared to that of other state-of-the-art randomizing-clock-frequency-based countermeasures. The comparison results are summarized in Table 5.3. Our countermeasure is tested on a RISC-V SoC integrated with an AES accelerator. A Kintex-7 FPGA is used to implement the entire system. Applying the proposed countermeasure results in the lowest area overhead, with an increase in LUT and FF of only 1.042 and 1.0055 times, respectively. The protected SoC has a 1.06-fold power cost and a 3.36-fold timing overhead, both of which are tolerable numbers. Resistance against power analysis attacks on the protected SoC is tested using the TVLA test, followed by practical CPA and DL-SCA attacks on all 16 bytes of the AES encryption key. With five million traces, the TVLA test is unable to detect any side-channel leakage from the protected SoC. In comparison to other relevant studies, this is the best TVLA result. Additionally, the RISC-V SoC equipped with our proposed countermeasure is impenetrable to typical CPA attacks at a trace count of five million. However, the protected SoC remains vulnerable to the profiled DL-SCA attacks. In our experiments, one million power traces are used in the profiling phase, and on one hundred thousand traces are used in the attacking phase. DL-SCA attacks are capable of revealing up to 13/16 subkeys.

However, as indicated in subsection 5.3.2.4, our proposed countermeasure contributes to DL-SCA resistance improvement under the identical profiling and attack settings.

Table 5.3: Comparison with related works.

	Targeted device	**Counter-measure**	**Overhead**			**TVLA**	**Attacked bytes**	**CPA**	**DL-SCA**	
			Timing	**Power**	**Area**				**Parameters**	**Results**
[51]	Multicore SoC (ASIC)	RTS, RIO, FPR, PSMC	1.04×	1.35×	1.23×	Failed with 200,000 traces	Only byte #0	2,000,000	Profiling: 1,000,000 Attacking: 100,000	Success on byte #0
[44]	RISC-V SoC (FPGA)	SCRIP	1.88×	N/A	LUT: 1.04× FF: 1.03× MMCM: 2×	Passed with 200,000 traces	N/A	300,000	N/A	
[45]	AES core (ASIC)	DFR-8	58.9×	1.01×	N/A	Passed with 5,000,000 traces	N/A	5,000,000	Profiling: 500,000 Attacking: N/A	Cannot reveal any byte
[43]	AES core (ASIC)	RFTC	1.72×	1.48×	1.3×	Passed with 1,000,000 traces	N/A	4,000,000	N/A	
This work	RISC-V SoC with AES accelerator (FPGA)	RDFS	3.36 ×	1.06×	LUT: 1.042× FF: 1.0055× MMCM: 2×	Passed with 5,000,000 traces	All 16 bytes	5,000,000	Profiling: 1,000,000 Attacking: 100,000	13/16 subkeys

Table 5.3 demonstrates that our protected design provides the highest level of resistance against CPA attacks. In the case of DL-SCA attacks, [45, 51] proposed better countermeasures that DL-SCA cannot overcome. Yang *et al.* proposed combining many countermeasures, including random task scheduling (RTS), random insertion of operations (RIO), frequency and phase randomization (FPR), and power state monitoring and control (PSMC) [51]. However, they only perform the security assessment by targeting the first byte of the secret key. The DL-SCA attacks failed to expose that first byte. Moreover, if each countermeasure is applied separately, the DL-SCA still successfully reveals the secret subkey. Hettwer *et al.* also use the MMCM to dynamically adjust the clock frequency of an AES core in [45]. While DL-SCA attacks are incapable of defeating their countermeasure, the timing overhead is considerable. Their proposed countermeasure increases the encryption time of the targeted AES core by roughly 59 times.

The experimental results mentioned before demonstrate the effectiveness of our proposed countermeasures against CPA and DL-SCA attacks in practice. The countermeasure generates a variety of various clock frequencies for operating the AES accelerator using the MMCM primitive in Xilinx's FPGA. Each encryption/decryption operation is carried out by the AES accelerator at a randomly chosen clock frequency. As a result, resistance against power analysis attacks is greatly increased while maintaining low overheads.

5.4 Further Evaluations

5.4.1 Applying rdfs into asic

In prior security evaluation of the proposed RDFS countermeasure, the target SoC is implemented on a Xilinx FPGA. Experimental results showed that the RDFS countermeasure is a simple-yet-effective technique, while the performance and resources penalty is relatively low. It is important to note that this countermeasure is a generic technique. the concept of RDFS countermeasure can also be applied to any other SoC implemented in ASIC. Therefore, in this section, an ASIC cryptographic SoC architecture is proposed to demonstrate this property of the proposed RDFS countermeasure.

5.4.1.1 Proposed architecture

The proposed system architecture of the cryptographic SoC is illustrated in Figure 5.15. The ASIC part is our target cryptographic SoC. It has all necessary components of a typical cryptographic SoC, including processing cores, a network of data buses, peripherals, *etc*. The proposed cryptographic SoC will also be used in future researches, hence, all SoC's components are selected as follows. The processing cores are two 64-bit RISC-V Rocket cores that support RV64GC extension of the RISC-V ISA, which means the proposed SoC supports the **I**nteger, **M**ultiplication, **A**tomic, single-precision (**F**), double-precision (**D**) floating-point, and **C**ompress extensions. Each core has 4KB of Layer-1 data cache and 4KB of instruction caches. 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 [78]. The Debug Unit, the Platform-level Interrupt Controller, the Core-Local Interrupts, and the Boot ROM are attached to the Control Bus. The Peripheral Bus is connected with standard communication peripherals, such as the Serial Peripheral Interface (SPI), the Universal asynchronous receiver-transmitter (UART), General Purpose Input-Output modules. A True Random Number Generator (TRNG) circuit is also attached to the Peripheral Bus. The TRNG circuit is used to produce random number, which will be used in various cryptographic operations. More-

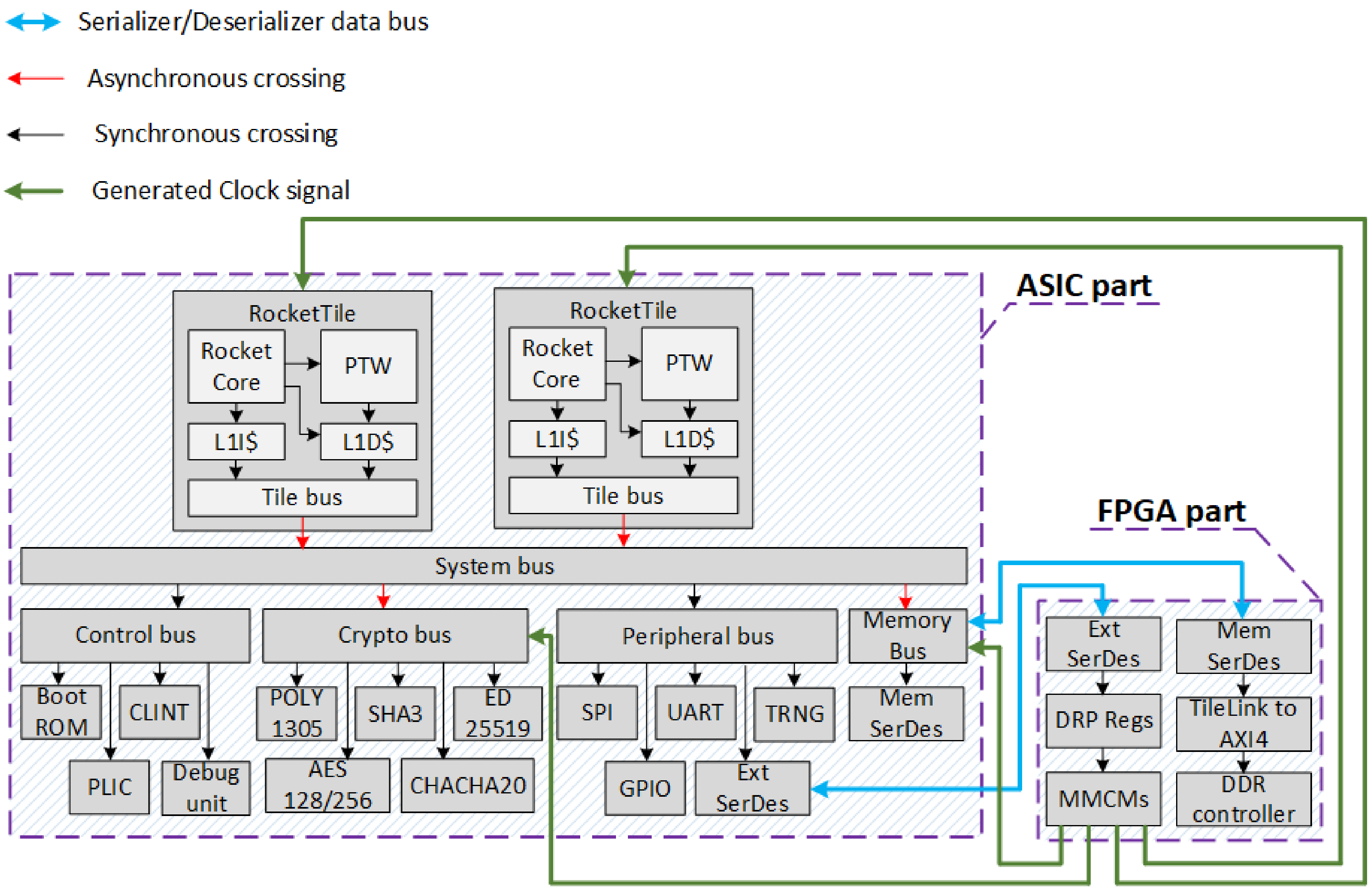


Figure 5.15: System architecture of the clock separated cryptographic SoC.

over, the Crypto Bus is used to attach all cryptographic hardware accelerators, including AES128/256, SHA3, ED25519, POLY1305, and CHACHA20. About the DDR controller and the MMCMs, since they are Xilinx IPs, they cannot be integrated into ASIC. Therefore, DDR controller and the MMCMs as well as other related modules, such as the Dynamic Reconfiguration Peripheral and the TileLink-to-AXI4 converter, are placed in an off-chip FPGA. Two pairs of Serializer/Deserializer (SerDes) converters are used to reduce the number of I/O pins required for the connections between the FPGA's modules and corresponding Buses (the Peripheral Bus and the Memory Bus). Besides, the several Tilelink connections are changed to asynchronous crossing to allows different parts of the proposed SoC can operate in different clock domains. The Tilelink connections that need to be modified are the connection between each RocketTile and the System Bus, the connection between the System Bus and the Crypto Bus, and the connection between the System Bus and the Memory Bus. As a result, each of two Rocket core, the group of all cryptographic accelerators, and the memory controller can be drove by separated clock signals. These separated clock signals are generated by the MMCMs integrated in the off-chip FPGA. These generated clock's frequencies can be fully controlled from software programs executed by processing cores.

5.4.1.2 FPGA Implementation

Two Xilinx VC707 Evaluation Boards are used to check the functionality of the proposed SoC. The test set up is described in Figure 5.16. The ASIC part of the proposed SoC is implemented on the Virtex-7 FPGA of one VC707 board, while the FPGA part is implemented on the other VC707 Board. Two FMC-to-GPIO extension cards are used extend the I/O pin of the Virtex-7 FPGAs. The SerDes connections is wired up using ribbon cables between these two FMC-to-GPIO extension cards. The generated clock signals is passed from FPGA part to ASIC part via SMA cables.

The proposed cryptographic SoC is implemented into the functional testing platform. The system booted successfully. The overall system operated as expected. All cryptographic accelerators produced correct encryption/decryption outputs. However, the maximum frequency of the system is only 20MHz, since connections using ribbon cable did not have the best quality.

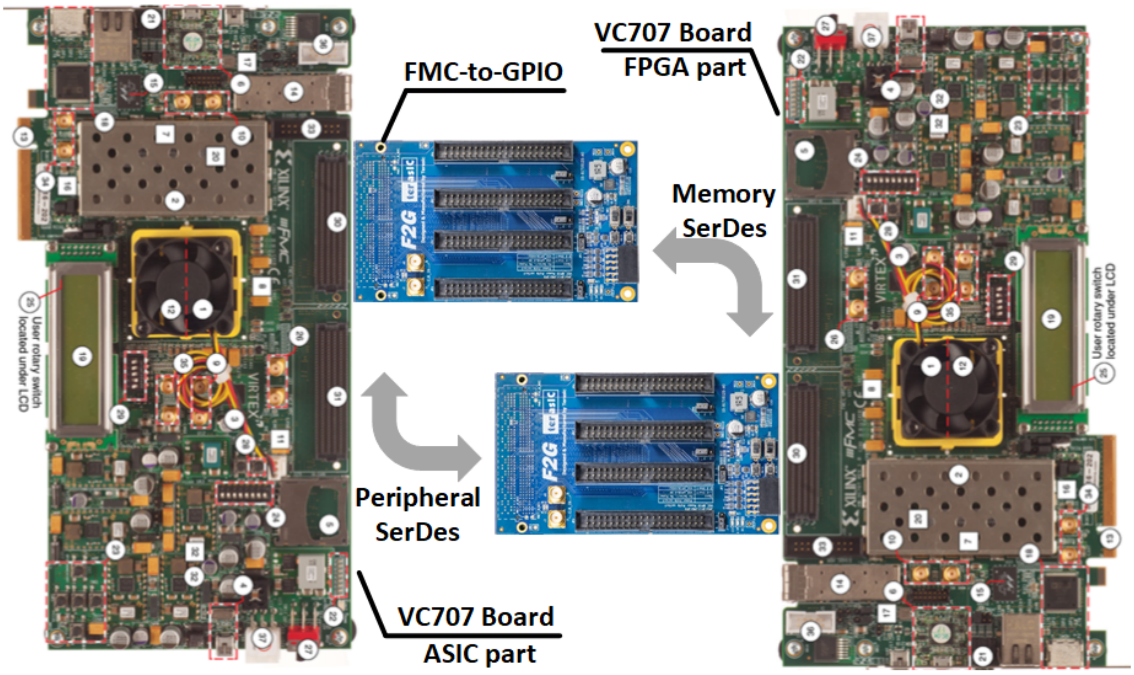


Figure 5.16: Testing functionality of the proposed clock separated cryptographic SoC.

5.4.1.3 VLSI Implementation

The ASIC part is implemented in VLSI circuit, using the ROHM 180nm technology. The layout of the chip is showed in Figure 5.17. The implemented chip has the size of 4,590-μm^2, which occupies the whole available area of the 5×5-mm^2 die. It can be seen in Figure 5.17 that the RV64GC Rocket core has the largest size, which is around 27% of whole die area. The second largest module is the ED25519 accelerator. Its size is larger than that of all other four cryptographic accelerators combined.

This layout has passed all the design rule checks, layout versus schematic checks, and was submitted for fabrication in early September of 2021. The final IC is delivered in December of 2021. A PCB is built for testing the fabricated IC. The PCB is an extension board, which would be placed on top of an FMC-to-GPIO board. Then, the FMC-to-GPIO board would be placed on the FMC1 connector of the VC707 FPGA board. The Virtex-7 FPGA chip on the VC707 board will contain the FPGA part of the overall system, including all three MMCMs and a DDR controller. The test PCB also hosts several headers for using external devices, such as the UART module, SPI module for Flash memory, SPI for SD card, JTAG debugger, input switches, and output LEDs. The overall test platform is shown in Figure 5.18.

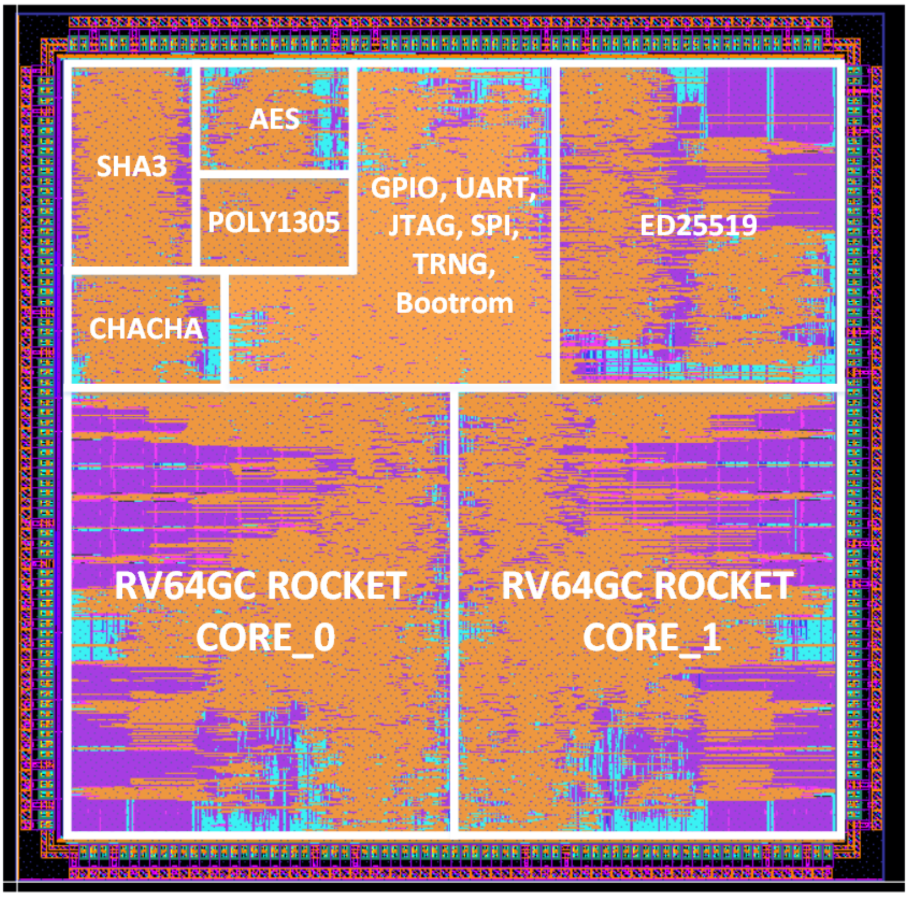


Figure 5.17: Chip layout of the ASIC part in ROHM 180nm technology.

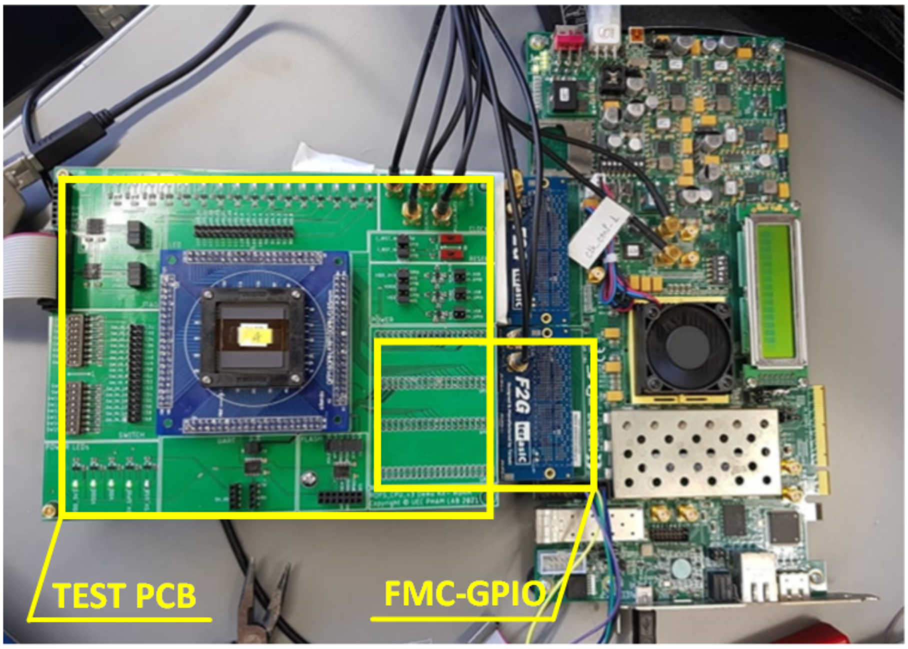
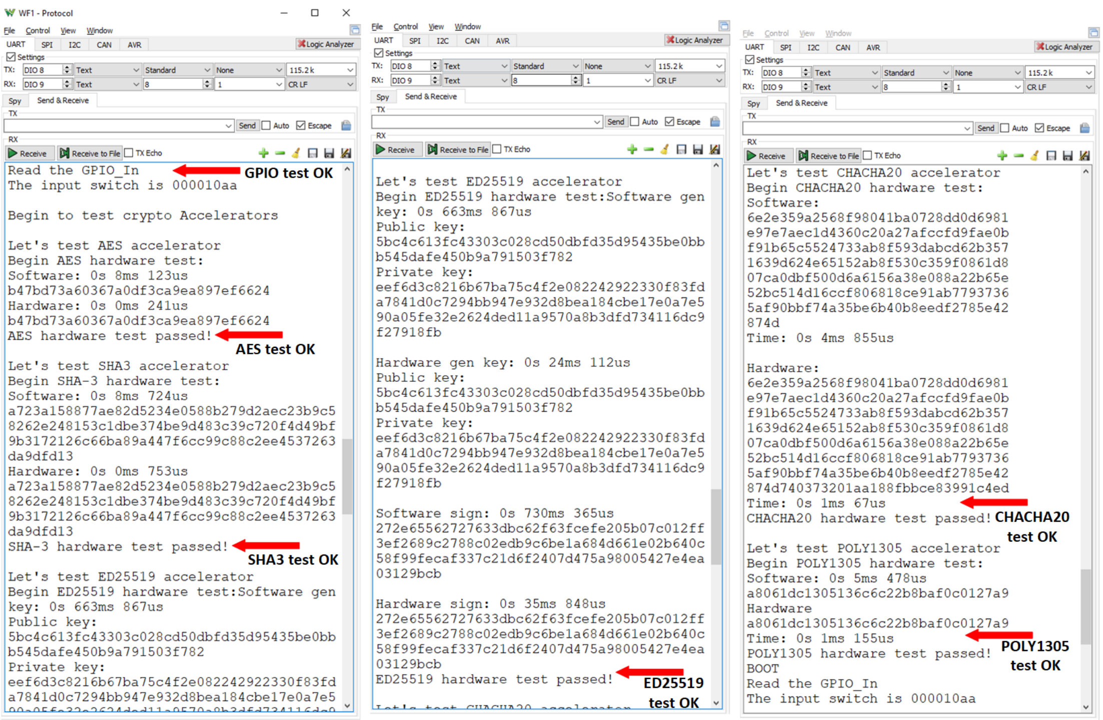


Figure 5.18: Test platform for the ASIC implementation of RDFS countermeasure in ROHM 180nm technology.

The ASIC chip is powered by external power sources. A test program, which includes functional tests for all cryptographic accelerators, is written and put into the external Flash memory. The ASIC chip will use the SPI port to read and then execute the test program. The test program contains software implementations of related cryptographic algorithms. These implementations are executed by the RISC-V cores. The outputs of these software implementations are then compared with the outputs of hardware accelerators. The results show that these outputs are matched, indicating that the ASIC chip is working correctly. Figure 5.19 shows the UART terminal output of used for testing all hardware crypto-accelerators.



☑ Settings
TX: DIO 8 | Text | Standard | None | 115.2 k
RX: DIO 9 | Text | 8 | 1 | CR LF
□ Auto ☑ Escape □ TX Echo

```
Read the GPIO_In
The input switch is 000010aa

Begin to test crypto Accelerators

Let's test AES accelerator
Begin AES hardware test:
Software: 0s 8ms 123us
b47bd73a60367a0df3ca9ea897ef6624
Hardware: 0s 0ms 241us
b47bd73a60367a0df3ca9ea897ef6624
AES hardware test passed!

Let's test SHA3 accelerator
Begin SHA-3 hardware test:
Software: 0s 8ms 724us
a723a158877ae82d5234e0588b279d2aec23b9c5
8262e248153c1dbe374be9d483c39c720f4d49bf
9b3172126c66ba89a447f6cc99c88c2ee4537263
da9dfd13
Hardware: 0s 0ms 753us
a723a158877ae82d5234e0588b279d2aec23b9c5
8262e248153c1dbe374be9d483c39c720f4d49bf
9b3172126c66ba89a447f6cc99c88c2ee4537263
da9dfd13
SHA-3 hardware test passed!

Let's test ED25519 accelerator
Begin ED25519 hardware test:Software gen
key: 0s 663ms 867us
Public key:
5bc4c613fc43303c028cd50dbfd35d95435be0bb
b545dafe450b9a791503f782
Private key:
eef6d3c8216b67ba75c4f2e082242922330f83fd
a7841d0c7294bb947e932d8bea184cbe17e0a7e5
```

```
Let's test ED25519 accelerator
Begin ED25519 hardware test:Software gen
key: 0s 663ms 867us
Public key:
5bc4c613fc43303c028cd50dbfd35d95435be0bb
b545dafe450b9a791503f782
Private key:
eef6d3c8216b67ba75c4f2e082242922330f83fd
a7841d0c7294bb947e932d8bea184cbe17e0a7e5
90a05fe32e2624ded11a9570a8b3dfd734116dc9
f27918fb

Hardware gen key: 0s 24ms 112us
Public key:
5bc4c613fc43303c028cd50dbfd35d95435be0bb
b545dafe450b9a791503f782
Private key:
eef6d3c8216b67ba75c4f2e082242922330f83fd
a7841d0c7294bb947e932d8bea184cbe17e0a7e5
90a05fe32e2624ded11a9570a8b3dfd734116dc9
f27918fb

Software sign: 0s 730ms 365us
272e65562727633dbc62f63fcefe205b07c012ff
3ef2689c2788c02edb9c6be1a684d661e02b640c
58f99fecaf337c21d6f2407d475a98005427e4ea
03129bcb

Hardware sign: 0s 35ms 848us
272e65562727633dbc62f63fcefe205b07c012ff
3ef2689c2788c02edb9c6be1a684d661e02b640c
58f99fecaf337c21d6f2407d475a98005427e4ea
03129bcb
ED25519 hardware test passed!

Let's test CHACHA20 accelerator
```

```
Let's test CHACHA20 accelerator
Begin CHACHA20 hardware test:
Software:
6e2e359a2568f98041ba0728dd0d6981
e97e7aec1d4360c20a27afccfd9fae0b
f91b65c5524733ab8f593dabcd62b357
1639d624e65152ab8f530c359f0861d8
07ca0dbf500d6a6156a38e088a22b65e
52bc514d16ccf806818ce91ab7793736
5af90bbf74a35be6b40b8eedf2785e42
874d
Time: 0s 4ms 855us

Hardware:
6e2e359a2568f98041ba0728dd0d6981
e97e7aec1d4360c20a27afccfd9fae0b
f91b65c5524733ab8f593dabcd62b357
1639d624e65152ab8f530c359f0861d8
07ca0dbf500d6a6156a38e088a22b65e
52bc514d16ccf806818ce91ab7793736
5af90bbf74a35be6b40b8eedf2785e42
874d740373201aa188fbbce83991c4ed
Time: 0s 1ms 67us
CHACHA20 hardware test passed!

Let's test POLY1305 accelerator
Begin POLY1305 hardware test:
Software: 0s 5ms 478us
a8061dc1305136c6c22b8baf0c0127a9
Hardware
a8061dc1305136c6c22b8baf0c0127a9
Time: 0s 1ms 155us
POLY1305 hardware test passed!
BOOT
Read the GPIO_In
The input switch is 000010aa
```

Figure 5.19: Terminal shows that the outputs of AES, SHA3, ED25519, CHACHA20, POLY1305 hardware accelerators, and GPIO are correct.

The power consumption of the chip is measured at the maximum system clock frequency of 5MHz. The measured results are illustrated in Figure 5.20, Figure 5.21, and Figure 5.22. Figure 5.20 show the amplitude of current consumed by the ASIC core in active mode, idle mode, and sleep modes. The active mode is the scenario where a core voltage VDD in the range of 1V to 2V, all clock signals of 5MHz are applied to the chip, and the chip is executing an always-loop. The idle mode is the scenario where the supply voltage and clock signals are supplied, but the chip is halted. It did not executing any instruction. The sleep mode is the scenario where the clocks are gated, only core voltage is supplied. Measuring the current in active mode would indicate the total consumption of the chip. Meanwhile, measuring the current in sleep mode would show the leakage current of the chip. It can be seen that the total active current is significantly increased from 11.33mA to 24.2mA when the core voltage VDD is increased from 1V to 2V. Meanwhile, the leakage current is slightly increased from 0.79μA to 1.4μA.

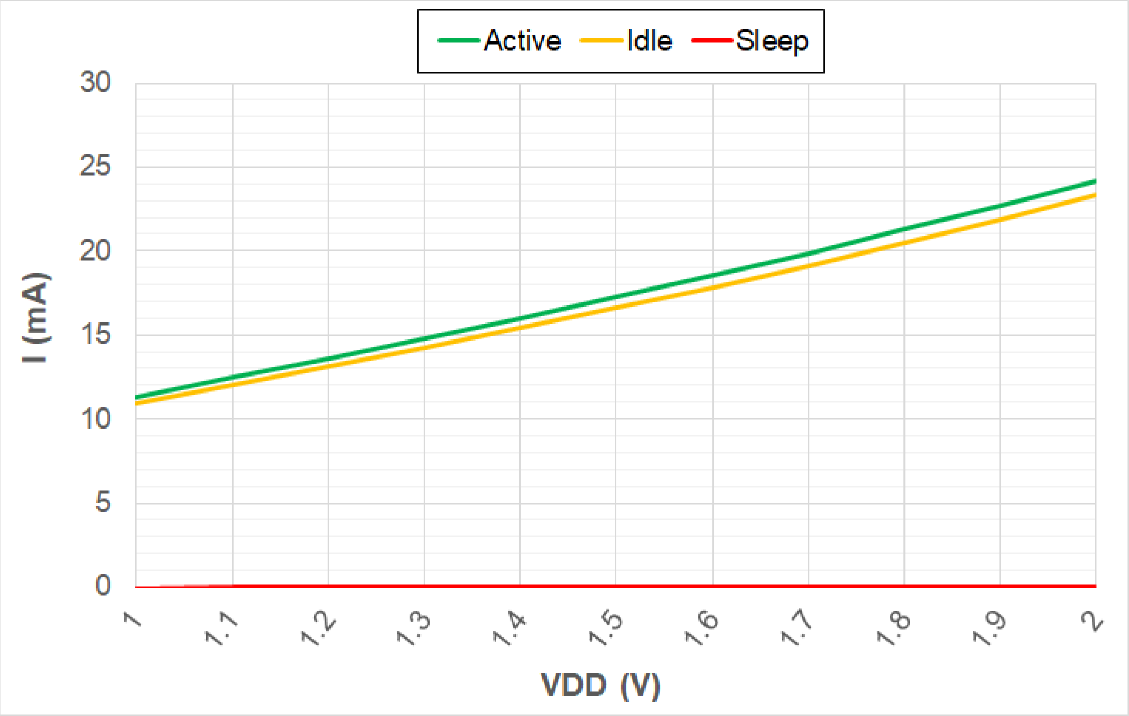


Figure 5.20: Current consumption in active and sleep mode, at different VDD.

With measured current consumption, the total power and leakage power of the ASIC chip can also be computed. The total power consumption increased from 11.33mW to 48.4mW when the core voltage VDD is increased from 1V to 2V. The leakage power also increased, but only from 0.79μW to 2.8μW. The ratio between the leakage power and the total power consumption can be described in Figure 5.21.

Finally, the energy consumed by the ASIC chip in nano Joule per clock cycle is computed and showed in Figure 5.22. In active mode, the chip requires around 2.23nJ/cycle to 9.68nJ/cycle, where the core voltage increases from 1V to 2V. Meanwhile, in the idle mode, the chip requires 2.178nJ/cycle to 9.336nJ/cycle. Further security evaluation will be set as future works.

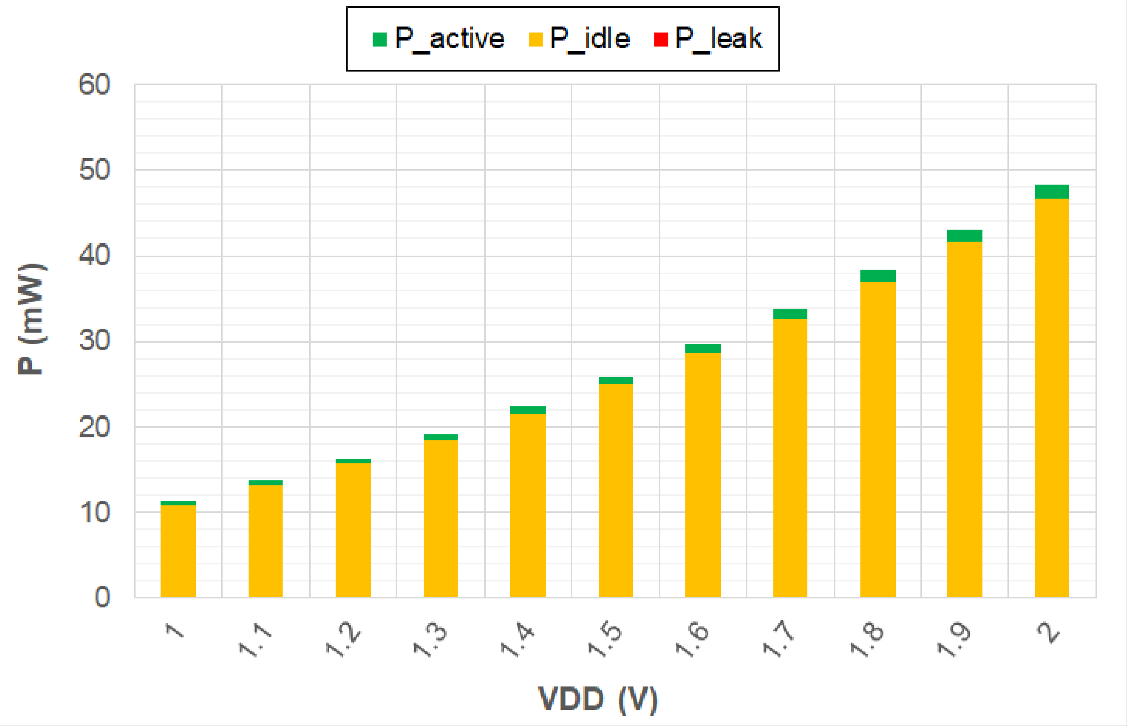


Figure 5.21: Power consumption in active, idle and sleep mode.

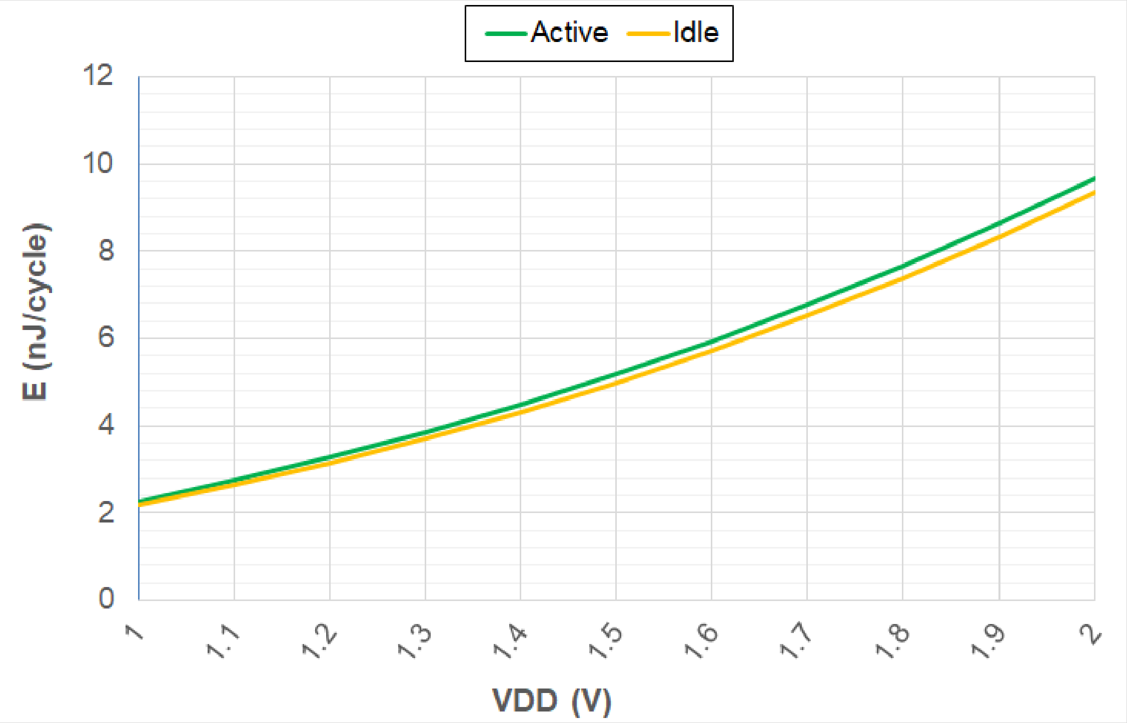


Figure 5.22: Energy consumption per clock cycle in active and sleep mode, at different VDD.

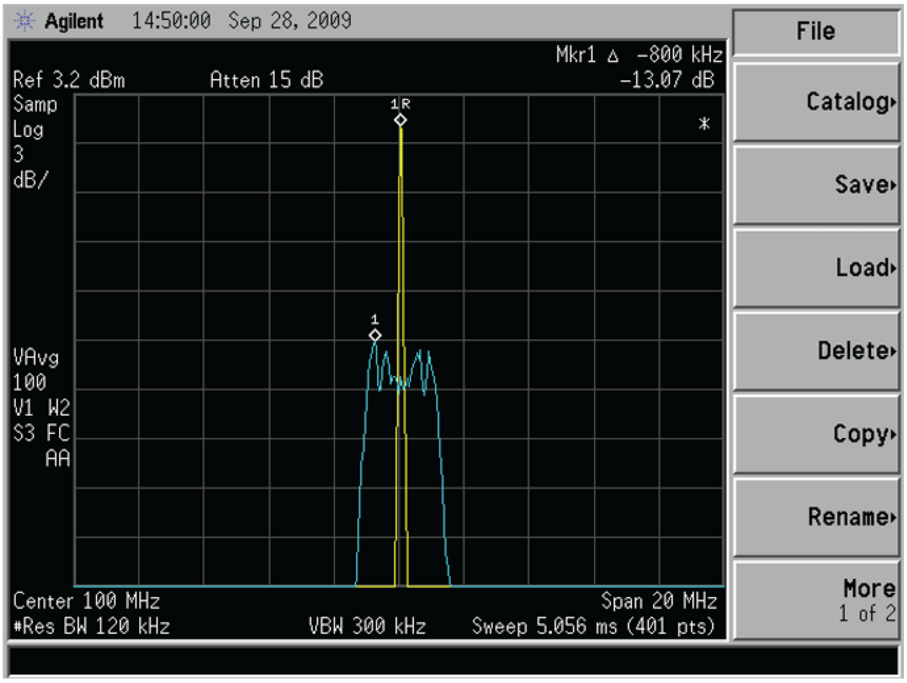
5.4.2 Spread spectrum

Spread Spectrum is a common technique that has been used in clock generation in some digital systems, such as microcontroller, micro processor, or Digital Signal Processor (DSP). Digital electronic devices such as microcontroller, micro processor, or Digital Signal Processor (DSP) often emit electromagnetic radiation that might interfere with the operation of other nearby devices. The electromagnetic interference (EMI) often has a very narrow frequency spectrum that consists of high amplitude peaks located at a specific frequency (equal to the main clock frequency) and is harmonics. The higher spectrum peaks cause more disturbances to adjacent devices. To regulate such interference, several techniques can be used, such as adding chokes, ferrite beads, or shielding. These technique could cause additional cost for developing the final product. A better solution is using the technique called Spread Spectrum clock generation. This technique helps to reduce the peak of electromagnetic interference of these digital system by slowly modulate the main clock signal in a variation range around the main clock signal. By doing so, the EMI's peaks are significantly reduced and the range of EMI's spectrum are widened.

Figure 5.23 demonstrates the effects of applying the Spead Spectrum technique to Xilinx Spartan-6 FPGAs [84]. The EMI spectrum caused by original design using fixed main clock signal of 100MHz is showed in yellow line. Meanwhile, the EMI spectrum caused by improved design with Spread Spectrum clock is showed in blue line. In this example, the 3.0% center-spread modulation mode is used. Figure 5.23 showed that by using the Spread Spectrum clock, the maximum energy of the EMI is reduced by 13dB and the power spectrum is wider than that of the original case.

In the context of power analysis attacks, the Spead Spectrum technique has some similarities with the proposed RDFS countermeasure in this dissertation. The instantaneous frequency of generated clock is dynamically changed during device's operation. Different clock frequencies cause different encrypting/decrypting times. Or in other words, Spread Spectrum can also cause misalignment in power consumption traces. Therefore, Spread Spectrum technique also can be used in countermeasures against power analysis attacks.

The proposed RDFS countermeasure is evaluated with the Xilinx Kintex-7 FPGA on the Sakura-X board. The driving clock signal of cryptographic core is generated by MMCM IPs. Fortunately, the MMCM IPs of Xilinx Kintex-7 FPGA also support the



Agilent 14:50:00 Sep 28, 2009

Mkr1 Δ -800 kHz
-13.07 dB

Ref 3.2 dBm Atten 15 dB
Samp
Log
3
dB/

VAvg
100
V1 W2
S3 FC
AA

Center 100 MHz Span 20 MHz
#Res BW 120 kHz VBW 300 kHz Sweep 5.056 ms (401 pts)

Figure 5.23: Effects of Center-Spread Modulation [84].

Spread Spectrum technique. According to the Production Guide [85], some important properties of Spread Spectrum technique in Kintex-7 FPGA can be listed as follow:

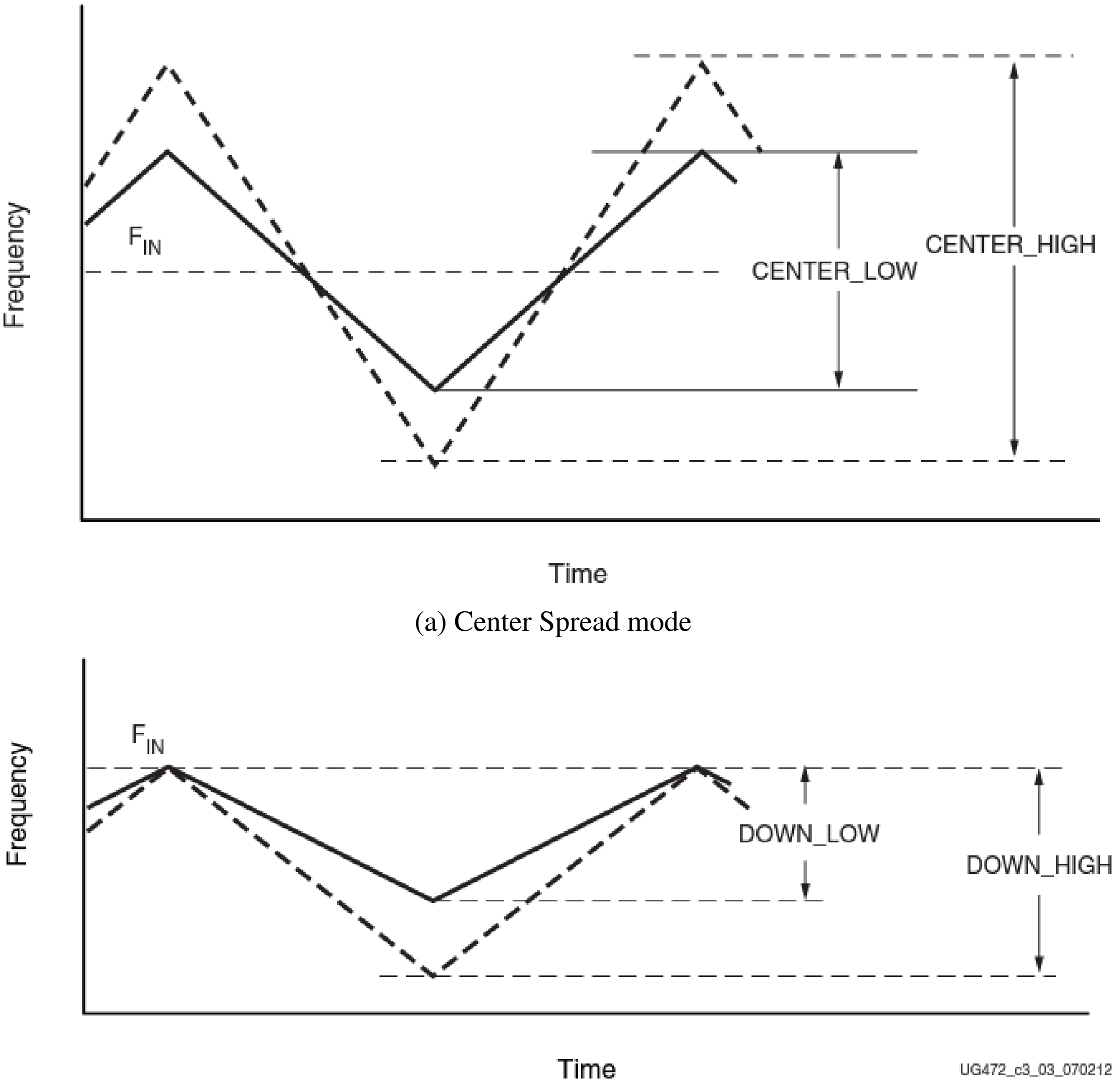
1. The features of Dynamic Re-configuring generated clock's frequency and phase are not available when the Spread Spectrum feature is selected.

2. The input frequency range is significantly reduced, from 10-800MHz to 25-150MHz.

3. CLK_OUT3 and CLK_OUT4 are no longer available.

4. There are four Spread Spectrum modes, including $DOWN\_LOW$ (down spread with low frequency spread), $DOWN\_HIGH$ (down spread with high frequency spread), $CENTER\_LOW$ (center spread with low frequency spread), $CENTER\_HIGH$ (center spread with high frequency spread). Figure 5.24 illustrates the differences between each mode. The modulation frequency can be selected in the range from 25KHz to 250KHz. The MMCM IP will compute the actual modulation period based on user-selected modulation frequency and the value of input clock frequency F_IN.

5. In each spread spectrum mode, certain constrains on the M, D, and O registers of each available output clock ports must be followed. In general, with each range



(a) Center Spread mode

(b) Down Spread mode

Figure 5.24: Spread Spectrum feature in Xilinx Kintex-7 FPGAs [85].

of input clock frequency F_IN, the the M, D, and O registers must take a certain value. Details of these constrains are provided in [81].

6. The frequency deviation of Spread-Spectrum clock is computed based on the value of the M, D, and O registers. In the best case ($CENTER\_HIGH$ mode), the largest frequency deviation is 2.44% of the input clock frequency, which is only around ± 3.66MHz.

It can be seen that the value of spread-spectrum clock can be changed in a very small range of frequency (around ± 3.66MHz). This range is much smaller than that of the proposed RDFS countermeasure, which is more than 200,000 distinct value in range from 50MHz to 100MHz. Therefore, using spread spectrum would create misalignment in power consumption traces but it will not be able to achieve the degree of misalignment offered by the RDFS countermeasure.

5.4.3 Frequency domain analysis

The experiment results in section 5.3.2 showed that the cryptographic SoC protected with RDFS-219,412 countermeasure can not be attacked by conventional CPA, even with five million traces. Each power trace will be a superposition of a waveform caused by overall system operating at 50MHz and a waveform caused by the target AES accelerator operating at a scaled clock frequency. Since the driving clock signal of the AES accelerator is proposed to be scaled to a different frequency after each encryption/decryption, each power trace will have a different pattern. A clever attacker could conduct Simple Power Analysis and aware about these changing pattern in collected power trace. Hence, he/she could perform pre-processing analysis in frequency domain before conducting actual CPA attacks. This section will discuss the trade-offs for attacker and provide a simple demonstration for such scenario.

5.4.3.1 Theoretical trade-offs

In [57], authors provided mathematical estimation of increment in number of power traces that are need for conventional CPA attacks to successfully attack on designs protected by hiding countermeasures. For direct CPA attacks on misaligned power traces (in time dimension), the attack results mainly depend on the probability $\hat{p}$, which is the maximum distribution of the POIs. Halving $\hat{p}$ will halve the correlation coefficient of the correct key guess and thus, quadruple the number of required power traces. For example, if there are 16 possible locations for a single POI in measured power traces, the $\hat{p}$ is $1/16$ and $16^2 = 256$ times more traces are needed.

In the general case, the proposed RDFS countermeasure can generate F distinct clock frequencies. Each frequency is used in one power traces. Assume that the randomness is good enough, there are F possible locations for the selected POI. It means that the probability $\hat{p}$ is $1/F$ and F^2 times more traces are needed.

If the attacker can perform frequency analysis on the power traces before conducting actual attacks, they might classify the power traces into several groups based on their frequency components. If the attacker can classify the power traces into G groups (each group contains power traces, which the AES accelerator frequencies belong to a consecutive range of frequency). Each group will have around F/G distinct frequencies. Hence, attacker now can attack on each group separately, and with the best group, he can obtain the key with only $(F/G)^2$ times more traces. However, he still need to measure the power traces of all other groups. Therefore the final trace increment is $(F/G)^2 \times G = F^2/G$ times.

The number of group G is less than or equal to the number of distinct frequencies F. They can only be equal if the frequency domain transformation is good enough and provide highest frequency resolution for the spectrum of the power traces. To do that, attacker will need a very good measuring equipment with very high sampling rate and a powerful computer to compute F-point Discrete Fourier Transform. We can assume that the attackers are capable of having those equipment. In such case, $F = G$, and the number of traces increment is $(F)^2/G = (F)^2/F = F$ times.

The proposed RDFS countermeasure in this chapter can generate 219,412 distinct clock frequencies. Each frequency is used in one power traces. Assume that the randomness is good enough, there are 219,412 possible locations for the selected POI. It means that the probability $\hat{p}$ is 219,412 and $219,412^2 \approx 48$ *billion* times more traces are needed. If attackers can perform trace classification with the best quality, then only $219,412$ times more traces are required. This is still a very good improvement.

5.4.3.2 Trace Classifying Demonstration

In this little experiment, the RDFS countermeasure using only two distinct frequencies are applied to the target cryptographic Soc for simplicity. Without loss of generality, these two frequencies are randomly picked to be around 57MHz and 80MHz. 200,000 power traces from the protected SoC are collected. The same measuring equipment are used. The Oscilloscope acquire the power traces with the sampling rate of 1GS/s. The length of each power trace is set to 1600 sample to cover entire encryption/decryption process. Each trace is transformed into frequency domain and classified into two separated groups based on its peaks in the spectrum.

Example of power traces with two different AES accelerator's clock frequencies are shown in Figure 5.25 and Figure 5.26 respectively.

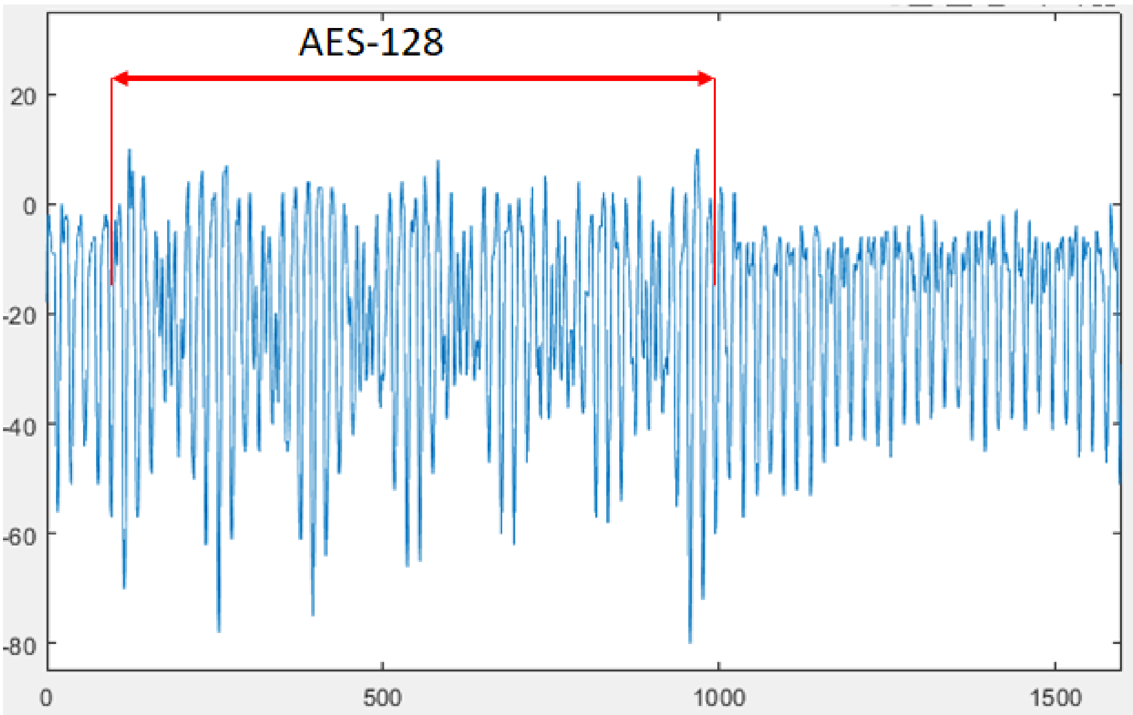


Figure 5.25: Example of power trace with 50MHz main CLK and 57MHz AES CLK.

Fourier transform is applied to each traces. In the spectrum, there are a highest peak corresponded to the main 50MHz clock at 50MHz location. There are also several lower harmonic peaks at 100MHz and 150MHz. Beside, there is a peak corresponded to the clock signal of the AES accelerator. Figure 5.27 shows the spectrum of example power traces.

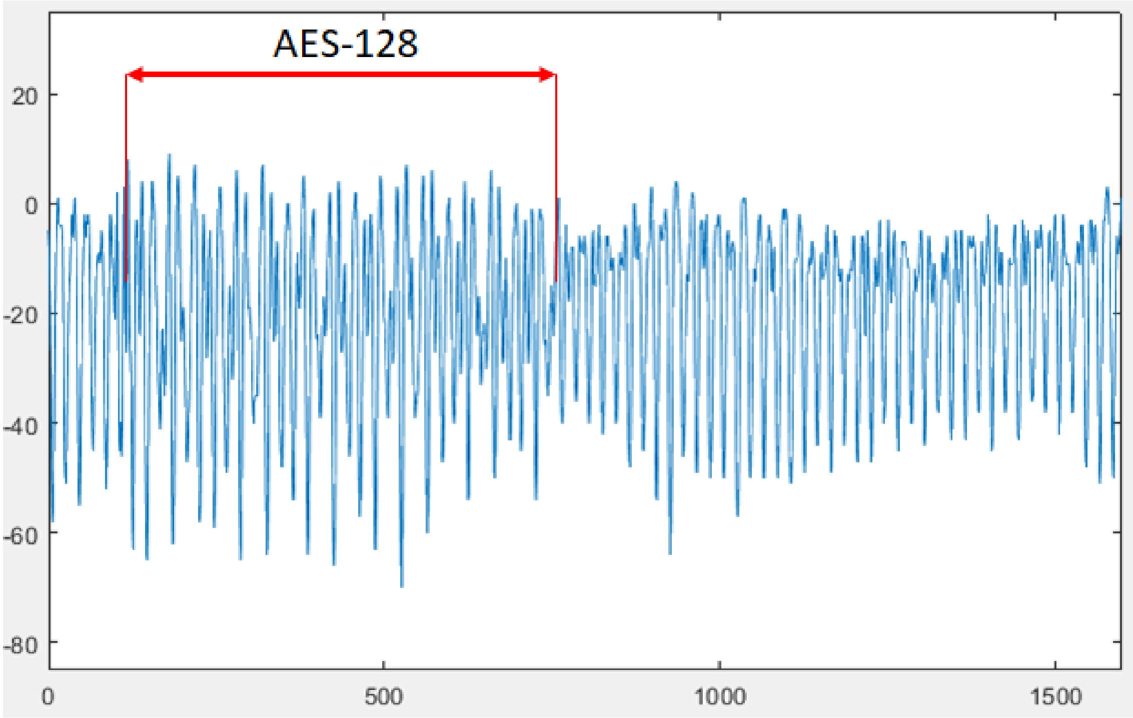


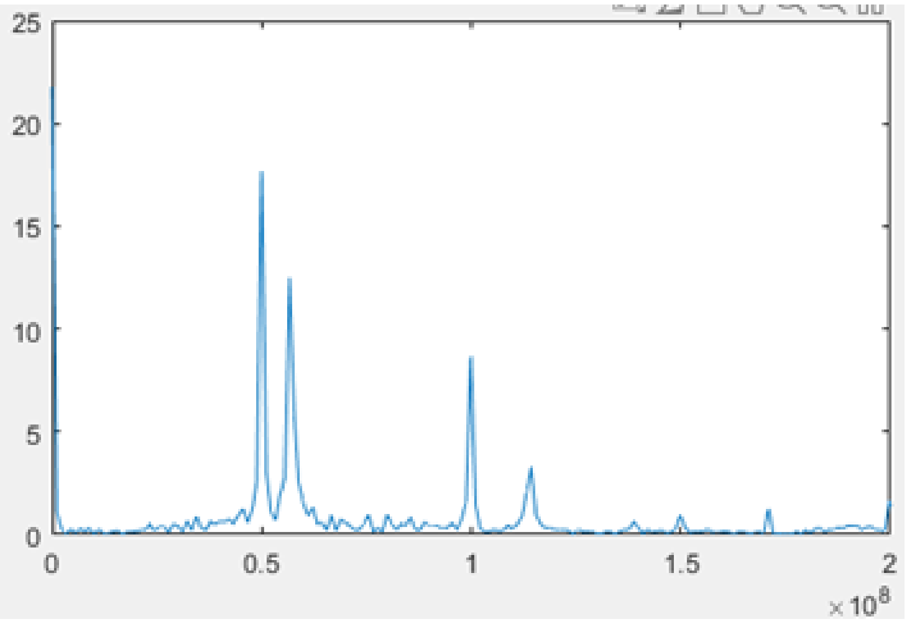
Figure 5.26: Example of power trace with 50MHz main CLK and 80MHz AES CLK.

Based on the location of these peaks, a MATLAB script is developed to automatically classify the trace. As a result, the first group contains 99,910 power traces that has AES's clock of 57MHz. The second group contains 100,078 power traces that has AES's clock of 80MHz.

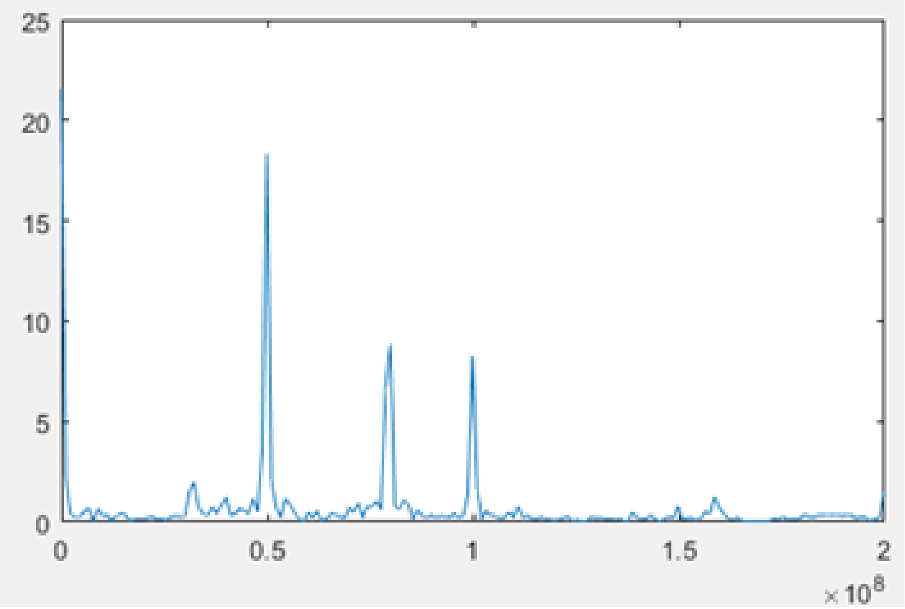
The CPA attack result on 200,000 mixed power traces is not good. Only 1/16 byte (the byte number 11) can be revealed. The PGE plot of that byte is illustrated in Figure 5.28. It showed that nearly 100,000 traces are required to reveal that single byte.

After that, CPA attacks are conducted on each classified group. For first group, the CPA attacks on 99,910 power traces successfully recover 03/16 bytes of the secret key (byte number 4, number 6, and number 11). The average number of required traces to reveal each correct byte is around 72,000 traces. For the second group, the CPA attacks on 100,078 power traces successfully recover 08/16 bytes of the secret key (byte number 1-4, and number 13-16). The average number of required traces to reveal each correct byte is around 50,750 traces. We can see that by classifying the power traces into two group, the number of required trace is halved in the best scenario. This result agrees with the theoretical estimation.

For further pre-processing the power traces, a suitable band-pass filter is applied to



(a) Spectrum of power trace with 50MHz main CLK and 57MHz AES CLK.



(b) Spectrum of power trace with 50MHz main CLK and 80MHz AES CLK.

Figure 5.27: Spectrum of example power traces.

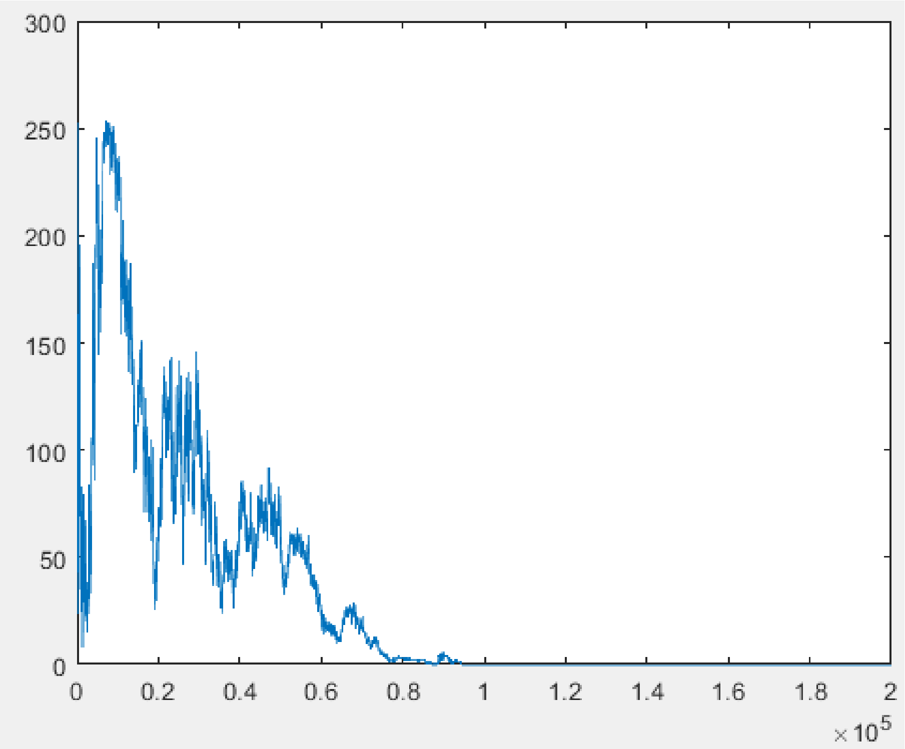


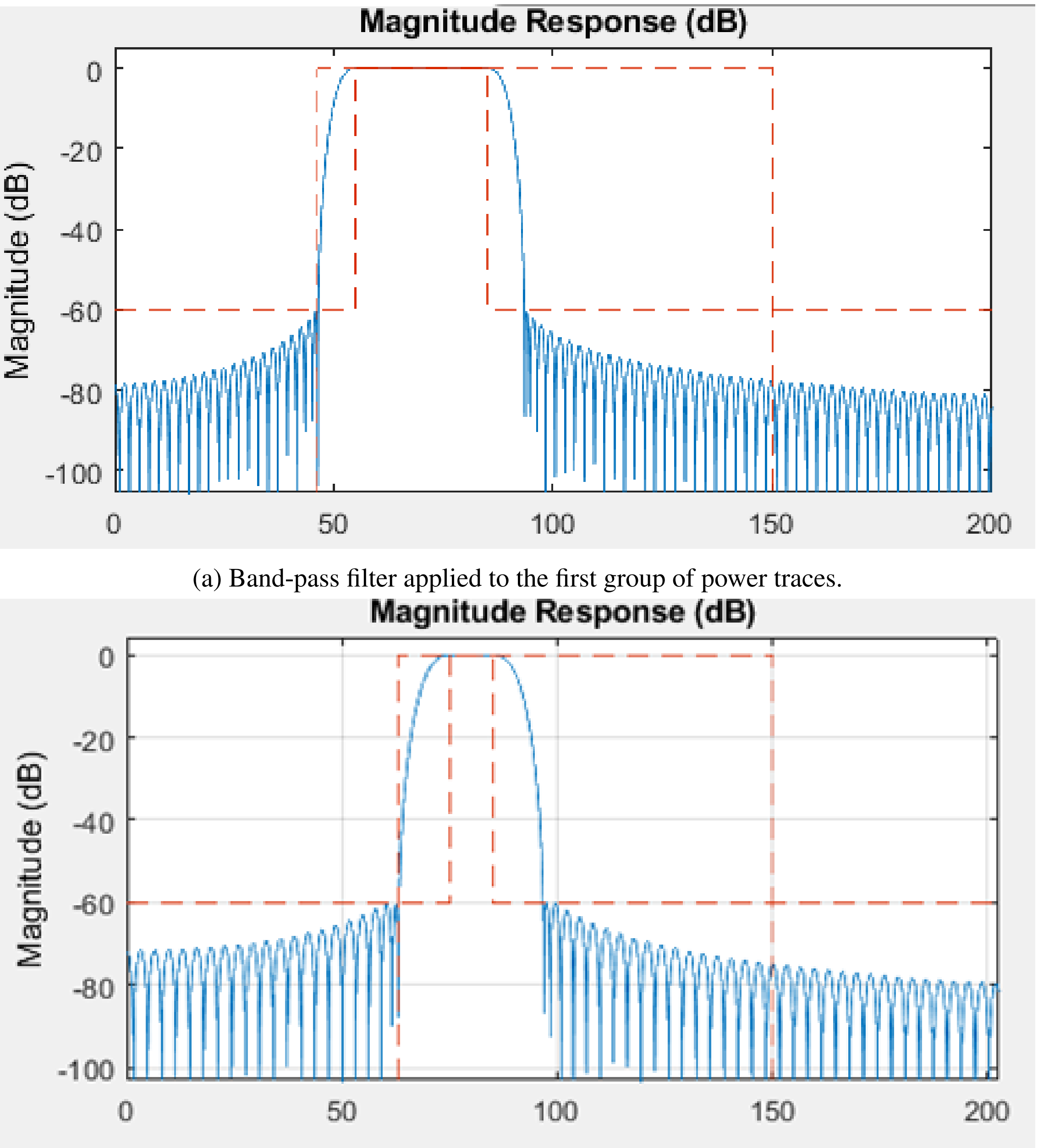
Figure 5.28: PGE of succeed byte when attacking on 200,000 mixed power traces.

each power traces to remove the spectrum of the main 50MHz clock and its harmonic peaks. The magnitude responses of two band-pass filters that have been used for two classified groups is showed in Figure 5.29.

As results, we have two new groups of power trace that only contain frequency components related to the AES's clock signal. The spectrum of each group is described in Figure 5.30. We can see that the 57MHz clock is quite close to the main clock of 50MHz, hence the peak corresponded to the main 50MHz clock cannot be completely removed. Then, CPA attacks are conducted on filtered traces. However the attack results is similar or even worse than attacking on unfiltered classified traces. For first group, the CPA attacks on 99,910 power traces successfully recover 03/16 bytes of the secret key. Meanwhile, for the second group, the CPA attacks on 100,078 power traces successfully recover only 05/16 bytes of the secret key.

5.5 Summary

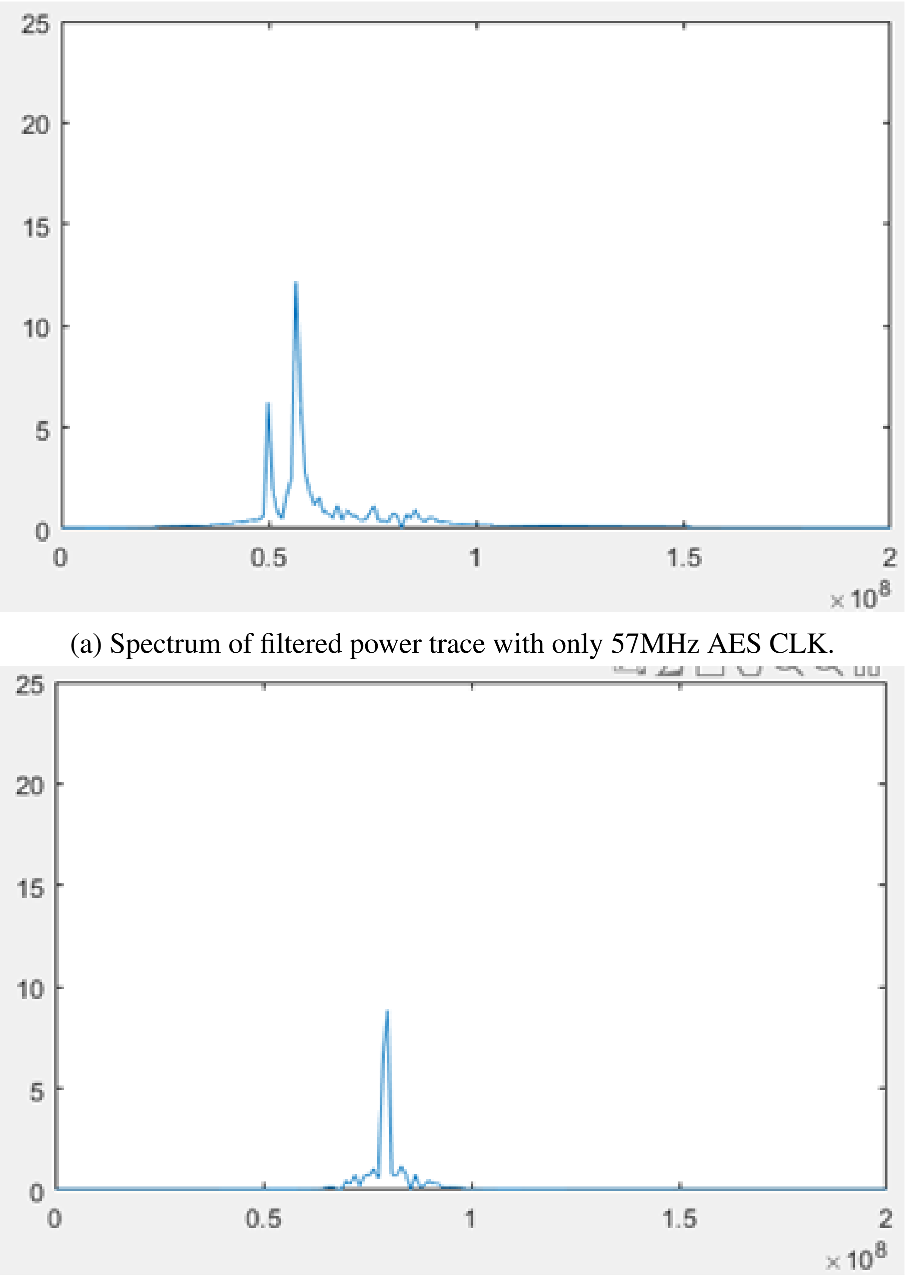
This chapter demonstrates the vulnerability of complex cryptographic SoCs to several types of power analysis attacks, even when the cryptographic components occupy



(a) Band-pass filter applied to the first group of power traces.

(b) Band-pass filter applied to the second group of power traces.

Figure 5.29: Magnitude response of the used band-pass filter.



(a) Spectrum of filtered power trace with only 57MHz AES CLK.

(b) Spectrum of filtered power trace with only 80MHz AES CLK.

Figure 5.30: Spectrum of example filtered power traces.

a small proportion of the overall system's hardware. Moreover, a hiding countermeasure against power analysis attacks is provided for such complex cryptographic SoCs. Practical CPA attacks, DL-SCA attacks, and TVLA leakage tests were used to determine its effectiveness. The experimental results demonstrate that the RISC-V SoC protected by the proposed countermeasure is invulnerable to conventional CPA attacks employing more than five million power traces, which is an improvement of at least 2,593 times. Additionally, the protected RISC-V SoC passed the TVLA leakage test with five million power traces, the highest number in comparison to other recent relevant works. Also, the proposed countermeasure aided in strengthening the targeted RISC-V SoC against advanced DL-SCA attacks.

Chapter 6

Conclusion

6.1 Achievements

Power analysis attacks are getting more and more popular when it comes to breaking security of cryptographic systems. It can be conducted using very basic measuring equipment and basic knowledge about the targeted system's architecture. Power consumption traces of the targeted system are measured by the attackers, while the targeted system is performing encryption/decryption using the secret key. Power analysis techniques are applied on the measured power traces to predict the used secret key. The first power analysis attack, the Simple Power Analysis attack, is introduced in 1999 [1]. Ever since, researchers have published numbers of more powerful power analysis attacks, which significantly increase the success rate of breaking the security of cryptographic systems. Several examples of these advanced power analysis attacks are Differential Power Analysis (DPA), Correlation Power Analysis (CPA), Deep Learning based Side Channel Attack (DLSCA). They are capable of overcome various countermeasures deployed at different levels of the cryptographic systems. Most of countermeasures are categorized as masking countermeasures or hiding countermeasures. Masking countermeasures use random values to obfuscate the real intermediate values being processed in the encryption/decryption. Meanwhile, hiding countermeasures aim to make the power consumption of the cryptographic devices independent of the intermediate values and independent of the operations that are performed. Even though masking techniques are capable of preventing SPA or first order DPA attacks, they are demonstrated to be vulnerable against high order DPA attacks and the state-of-the art DLSCA. Hiding countermeasures are considered as better solutions to prevent advanced power analysis attacks. This dissertation presents several novel hardware-based hiding countermeasures to enhance the power analysis resistance of different types of cryptographic devices. The key conclusion derived from this dissertation are:

- The first proposed hiding countermeasure in this dissertation aims to be applied for any cryptographic devices fabricated using FDSOI technology. It makes use of the back-gate biasing technique to enhance these device's resistance against power analysis attacks. Previously, the back-gate bias technique is only used to optimize the targeted device for high-performance (with forward back-gate bias) or low-power consumption (with reverse back-gate bias).

- Experiments provided in this dissertation show that using forward back-gate bias not only increase device's performance, it also improves device's resistance against power analysis attacks up to 14.5 times. The back-gate bias of the targeted cryptographic device can also be randomly changed in the forward bias region to further enhance the resistance against power analysis attacks up to 33.4 times, at a small cost of hardware overheads.

- This dissertation proposed a second hiding countermeasure named RDFS. It is designed to be applied in complex SoCs integrated with cryptographic hardware accelerators. The clock frequency driving the cryptographic hardware accelerators is randomly altered after each encryption/decryption, while the clock frequency driving the rest of the SoC is maintained unchanged.

- Effectiveness of the RDFS countermeasure is determined by conducting TVLA test, conventional CPA attacks and the state-of-the-art DLSCA attacks. Experimental results show that the RISC-V SoC protected by the proposed countermeasure is invulnerable to conventional CPA attacks employing more than five million power traces, which is an improvement of at least 2,593 times. Additionally, the protected RISC-V SoC passed the TVLA leakage test with five million power traces, the highest number in comparison to other recent relevant works. Also, the proposed countermeasure aided in strengthening the targeted RISC-V SoC against advanced DL-SCA attacks.

6.2 Limitations

This dissertation illustrated that the proposed countermeasures have decent effectiveness in resisting power analysis attacks. However, these proposals still contain several limitations as shown below:

- First, when evaluating the first proposed countermeasure, we employed an open-source, general-purpose microcontroller. There is no voltage regulator and no true random number generator integrated inside the target microcontroller. Therefore, off-chip components must be used to generate and apply random values for the back-gate bias and supply voltage. Furthermore, the pseudorandom values are generated by software, which is sufficient for evaluating effectiveness of the proposed countermeasures. However, in practice, it should be avoided because an attacker may simply take over the control or shut down the random number generator and voltage regulator, neutralizing the proposed countermeasures.

- Second, the back-gate bias control is applied to the entire logic core of the targeted microcontroller. When the proposed back-gate bias countermeasure are deployed, modules unrelated to the processing of cryptographic functions are also biased in the forward bias region. These blocks will generate an excessive amount of wasted leakage currents that will be ineffective in preventing DPA attacks.

- Third, When evaluating the second proposed countermeasure (RDFS), we employed a cryptographic RISC-V SoC inherited from prior work. The SoC includes only an LFSR-based PRNG as an MMIO peripheral. As a result, the random values used to determine the operating frequency of the AES accelerator are only pseudorandom. An integrated true random number generator would increase the randomness of the generated values, hence increasing resistance against power analysis attacks further more.

- Lastly, the PRNG is integrated as an independent peripheral. Hence, it adds additional time overhead, as random number generation must occur prior to creating the corresponding clock frequency and encryption. Having another dedicated random number generator would address the problem and minimize the time overhead even more.

- Lastly, Mixed Mode Clock Managers (MMCM) are used to generate different clock signals in the RDFS countermeasure. MMCM is an Intellectual Property design of Xilinx. Therefore, The RDFS countermeasure must be implemented on FPGA. If designers want to employ the RDFS countermeasure in ASIC, they must build their own Clock generator module.

6.3 Future Works

The future works will focus on eliminating the previously mentioned limitations and improving the proposed countermeasures. For the first proposed countermeasure (RDBB), future works will be directed toward integrating an on-chip true random number generator, a voltage regulator, and partial back-gate biasing into the targeted devices. This proposed countermeasure can also be considered as a noise-controlling-based countermeasure. Therefore, it will be interesting to test its effectiveness against the state-of-the-art DLSCA.

For the second proposed countermeasure (RDFS), the future works will be directed toward integrating an on-chip true random number generator, dedicating only for the countermeasure against power analysis attacks. Another possible task is designing an Clock generator module so the proposed countermeasure would be independent of any Intellectual Property. Based on the idea of trace classifying, a more advanced pre-processing method could be developed in the future to realign the power traces of the protected Cryptographic SoC. The attackers could first transform the power traces into the frequency domain to obtain information about the clock frequency of the AES accelerator. Then, they can select all the power traces that have approximately similar AES accelerator's clock frequencies, and use FFT and bandpass filter to filter out all unrelated frequency components. Finally, they we can apply a stretching-based realignment method (such as Dynamic Time Warp) instead of a conventional shifting-based method like pattern matching.

References

[1] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential Power Analysis”. In: *Advances in Cryptology — CRYPTO’ 99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. ISBN: 978-3-540-48405-9.

[2] Mehdi-Laurent Akkar, Régis Bevan, Paul Dischamp, and Didier Moyart. “Power Analysis, What Is Now Possible...” In: *Advances in Cryptology — ASIACRYPT 2000*. Ed. by Tatsuaki Okamoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 489–502. ISBN: 978-3-540-44448-0.

[3] Eric Brier, Christophe Clavier, and Francis Olivier. “Correlation Power Analysis with a Leakage Model”. In: *Cryptographic Hardware and Embedded Systems - CHES 2004*. Ed. by Marc Joye and Jean-Jacques Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29. ISBN: 978-3-540-28632-5.

[4] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. “Multi-channel Attacks”. In: *Cryptographic Hardware and Embedded Systems - CHES 2003*. Ed. by Colin D. Walter, Çetin K. Koç, and Christof Paar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 2–16. ISBN: 978-3-540-45238-6.

[5] Régis Bevan and Erik Knudsen. “Ways to Enhance Differential Power Analysis”. In: *Information Security and Cryptology — ICISC 2002*. Ed. by Pil Joong Lee and Chae Hoon Lim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 327–342. ISBN: 978-3-540-36552-5.

[6] X. Cai, R. Li, S. Kuang, and J. Tan. “An Energy Trace Compression Method for Differential Power Analysis Attack”. In: *IEEE Access* 8 (2020), pp. 89084–89092.

[7] J. Ai, Z. Wang, X. Zhou, and C. Ou. “Improved wavelet transform for noise reduction in power analysis attacks”. In: *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. 2016, pp. 602–606.

[8] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. “An AES Smart Card Implementation Resistant to Power Analysis Attacks”. In: *Applied Cryptography and Network Security*. Ed. by Jianying Zhou, Moti Yung, and Feng Bao. Berlin,

Heidelberg: Springer Berlin Heidelberg, 2006, pp. 239–252. ISBN: 978-3-540-34704-0.

[9] Yongdae Kim and Haengseok Ko. “Using Principal Component Analysis for Practical Biasing of Power Traces to Improve Power Analysis Attacks”. In: *Information Security and Cryptology – ICISC 2013*. Ed. by Hyang-Sook Lee and Dong-Guk Han. Cham: Springer International Publishing, 2014, pp. 109–120. ISBN: 978-3-319-12160-4.

[10] NewAE Technology Inc. *Tutorial B5 Breaking AES (Straightforward), Version 4*. 2017. URL: `http://wiki.newae.com/V4:Tutorial_B5_Breaking_AES_(Straightforward)`.

[11] R. Xu, L. Zhu, A. Wang, X. Du, K. R. Choo, G. Zhang, and K. Gai. “Side-Channel Attack on a Protected RFID Card”. In: *IEEE Access* 6 (2018), pp. 58395–58404.

[12] M. Zhao and G. E. Suh. “FPGA-Based Remote Power Side-Channel Attacks”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 229–244.

[13] W. Wang, Y. Yu, F. Standaert, J. Liu, Z. Guo, and D. Gu. “Ridge-Based DPA: Improvement of Differential Power Analysis For Nanoscale Chips”. In: *IEEE Transactions on Information Forensics and Security* 13.5 (2018), pp. 1301–1316.

[14] Shourong Hou, Yujie Zhou, Hongming Liu, and Nianhao Zhu. “Wavelet support vector machine algorithm in power analysis attacks”. In: *Radioengineering* 26.3 (2017), pp. 890–902.

[15] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. “Breaking cryptographic implementations using deep learning techniques”. In: *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer. 2016, pp. 3–26.

[16] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. “Convolutional neural networks with data augmentation against jitter-based countermeasures”. In: *International Conference on Cryptographic Hardware and Embedded Systems*. Springer. 2017, pp. 45–68.

[17] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. “Study of deep learning techniques for side-channel analysis and introduction to ASCAD database”. In: *ANSSI, France & CEA, LETI, MINATEC Campus, France. Online verfügbar unter https://eprint. iacr. org/2018/053. pdf, zuletzt geprüft am* 22 (2018), p. 2018.

[18] Li Duan, Zhang Hongxin, Li Qiang, Zhao Xinjie, and He Pengfei. “Electromagnetic side-channel attack based on PSO directed acyclic graph SVM”. In: *The Journal of China Universities of Posts and Telecommunications* 22.5 (2015), pp. 10–15.

[19] Ahmed Mahmoud, Ulrich Rührmair, Mehrdad Majzoobi, and Farinaz Koushanfar. “Combined Modeling and Side Channel Attacks on Strong PUFs.” In: *IACR Cryptol. ePrint Arch.* 2013 (2013), p. 632.

[20] Benjamin Timon. “Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.2 (Feb. 2019), pp. 107–131. DOI: `10.13154/tches.v2019.i2.107-131`. URL: `https://tches.iacr.org/index.php/TCHES/article/view/7387`.

[21] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. “Template Attacks”. In: *Cryptographic Hardware and Embedded Systems - CHES 2002*. Ed. by Burton S. Kaliski, çetin K. Koç, and Christof Paar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 13–28. ISBN: 978-3-540-36400-9.

[22] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. “Univariate side channel attacks and leakage modeling”. In: *Journal of Cryptographic Engineering* 1.2 (2011), p. 123.

[23] Abhishek Chakraborty, Bodhisatwa Mazumdar, and Debdeep Mukhopadhyay. “A practical dpa on grain v1 using ls-svm”. In: *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE. 2015, pp. 44–47.

[24] Shourong Hou, Yujie Zhou, Hongming Liu, and Nianhao Zhu. “Wavelet support vector machine algorithm in power analysis attacks”. In: *Radioengineering* 26.3 (2017), pp. 890–902.

[25] Mehdi-Laurent Akkar and Louis Goubin. “A Generic Protection against High-Order Differential Power Analysis”. In: *Fast Software Encryption*. Ed. by Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 192–205. ISBN: 978-3-540-39887-5.

[26] Thomas S. Messerges. “Securing the AES Finalists Against Power Analysis Attacks”. In: *Fast Software Encryption*. Ed. by Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 150–164. ISBN: 978-3-540-44706-1.

[27] Yvo Desmedt. “Some recent research aspects of threshold cryptography”. In: *Information Security*. Ed. by Eiji Okamoto, George Davida, and Masahiro Mambo. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 158–173. ISBN: 978-3-540-69767-1.

[28] E. D. Mulder, S. Gummalla, and M. Hutter. “INVITED: Protecting RISC-V against Side-Channel Attacks”. In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*. 2019, pp. 1–4.

[29] L. Benini, E. Omerbegovic, A. Macii, M. Poncino, E. Macii, and F. Pro. “Energy-aware design techniques for differential power analysis protection”. In: *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*. 2003, pp. 36–41.

[30] E. Laohavaleeson and C. Patel. “Current flattening circuit for DPA countermeasure”. In: *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. 2010, pp. 118–123.

[31] 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: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 142–143. DOI: 10.1109/ISSCC.2017.7870301.

[32] Shengqi Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Yuan Xie. “Power attack resistant cryptosystem design: a dynamic voltage and frequency switching approach”. In: *Design, Automation and Test in Europe*. 2005, 64–69 Vol. 3.

[33] 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: *Proceedings of the 28th European Solid-State Circuits Conference*. 2002, pp. 403–406.

[34] Kris Tiri and Ingrid Verbauwhede. “Secure Logic Synthesis”. In: *Field Programmable Logic and Application*. Ed. by Jürgen Becker, Marco Platzner, and Serge Vernalde. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1052–1056. ISBN: 978-3-540-30117-2.

[35] P. E. Andrews and M. S. Dhanesh. “A body biased adiabatic dynamic differential logic(BADDL) to prevent DPA attacks in smart cards”. In: *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2017, pp. 686–690.

[36] A. Alipour, A. Papadimitriou, V. Beroulle, E. Aerabi, and D. Hély. “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: *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2020, pp. 614–617. DOI: 10.23919/DATE48585.2020.9116387.

[37] Shanggong Feng, Junning Wu, Shengang Zhou, and Renwei Li. “The Implementation of LeNet-5 with NVDLA on RISC-V SoC”. In: *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. 2019, pp. 39–42. DOI: 10.1109/ICSESS47205.2019.9040769.

[38] Xinchao Zhong, Chiu-Wing Sham, and Longyu Ma. “A RISC-V SoC for Mobile Payment Based on Visible Light Communication”. In: *2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. 2020, pp. 102–105. DOI: 10.1109/APCCAS50809.2020.9301688.

[39] P. Flatresse, G. Cesana and X. Cauchy. *Planar fully depleted silicon technology to design competitive SOC at 28nm and beyond*. February, 2012. URL: %7Bhttps://www.soitec.com/media/documents/5/file/planar_fd_silicon_technology_competitive_soc_28nm.pdf%7D.

[40] T. Ishigaki, R. Tsuchiya, Y. Morita, H. Yoshimoto, N. Sugii, T. Iwamatsu, H. Oda, Y. Inoue, T. Ohtou, T. Hiramoto, and S. Kimura. “Silicon on thin BOX (SOTB) CMOS for ultralow standby power with forward-biasing performance booster”. In: *ESSDERC 2008 - 38th European Solid-State Device Research Conference*. 2008, pp. 198–201.

[41] Trong-Thuc Hoang, Ckristian Duran, Khai-Duy Nguyen, Tuan-Kiet Dang, Quynh Nguyen Quang Nhu, Phuc Hong Than, Xuan-Tu Tran, Duc-Hung Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham. “Low-power high-performance 32-bit RISC-V microcontroller on 65-nm silicon-on-thin-BOX (SOTB)”. In: *IEICE Electronics Express* 17.20 (2020), pp. 20200282–20200282. DOI: 10.1587/elex.17.20200282.

[42] Tim Güneysu and Amir Moradi. “Generic side-channel countermeasures for reconfigurable devices”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2011, pp. 33–48.

[43] Darshana Jayasinghe, Aleksandar Ignjatovic, and Sri Parameswaran. “RFTC: Runtime Frequency Tuning Countermeasure Using FPGA Dynamic Reconfiguration to Mitigate Power Analysis Attacks”. In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*.

[44] Darshana Jayasinghe, Aleksandar Ignjatovic, and Sri Parameswaran. “SCRIP: Secure Random Clock Execution on Soft Processor Systems to Mitigate Power-based Side Channel Attacks”. In: *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2019, pp. 1–7. DOI: 10.1109/ICCAD45719.2019.8942112.

[45] Benjamin Hettwer, Kallyan Das, Sebastien Leger, Stefan Gehrer, and Tim Güneysu. “Lightweight Side-Channel Protection using Dynamic Clock Randomization”. In: *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*. 2020, pp. 200–207. DOI: 10.1109/FPL50879.2020.00041.

[46] D. Das, S. Maity, S. B. Nasir, S. Ghosh, A. Raychowdhury, and S. Sen. “ASNI: Attenuated Signature Noise Injection for Low-Overhead Power Side-Channel Attack Immunity”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.10 (2018), pp. 3300–3311. DOI: 10.1109/TCSI.2018.2819499.

[47] K. Baddam and M. Zwolinski. “Evaluation of Dynamic Voltage and Frequency Scaling as a Differential Power Analysis Countermeasure”. In: *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID’07)*. 2007, pp. 854–862.

[48] A. Singh, M. Kar, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay. “25.3 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: *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. 2019, pp. 404–406. DOI: 10.1109/ISSCC.2019.8662344.

[49] Martin Petrvalsky and Milos Drutarovsky. “Constant-weight coding based software implementation of DPA countermeasure in embedded microcontroller”. In: *Microprocessors and Microsystems* 47 (2016), pp. 82–89.

[50] HanBit Kim, HeeSeok Kim, and Seokhie Hong. “Power-Balancing Software Implementation to Mitigate Side-Channel Attacks without Using Look-Up Tables”. In: *Applied Sciences* 10.7 (2020), p. 2454.

[51] J. Yang, J. Han, F. Dai, W. Wang, and X. Zeng. “A Power Analysis Attack Resistant Multicore Platform With Effective Randomization Techniques”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.6 (2020), pp. 1423–1434. DOI: 10.1109/TVLSI.2020.2971636.

[52] Ali A El-Moursy, Abdollah M Darya, Ahmed S Elwakil, Abhinand Jha, and Sohaib Majzoub. “Chaotic Clock Driven Cryptographic Chip: Towards a DPA Resistant AES Processor”. In: *IEEE Transactions on Emerging Topics in Computing* (2020).

[53] Jean-Sébastien Coron and Ilya Kizhvatov. “An efficient method for random delay generation in embedded software”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2009, pp. 156–170.

[54] Michael Tunstall and Olivier Benoit. “Efficient use of random delays in embedded software”. In: *IFIP International Workshop on Information Security Theory and Practices*. Springer. 2007, pp. 27–38.

[55] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. “An AES smart card implementation resistant to power analysis attacks”. In: *International conference on applied cryptography and network security*. Springer. 2006, pp. 239–252.

[56] Kazuo Sakiyama, Yu Sasaki, and Yang Li. *Security of block cipher: From algorithm design to hardware implementation. From algorithm design to hardware implementation*. Wiley, 2015.

[57] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards. Revealing the Secrets of Smart Cards*. Springer, 2007.

[58] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. "A testing methodology for side-channel resistance validation". In: *NIST non-invasive attack testing workshop*. Vol. 7. 2011, pp. 115–136.

[59] Jeremy Cooper, Elke DeMulder, Gilbert Goodwill, Joshua Jaffe, Gary Kenworthy, Pankaj Rohatgi, et al. "Test vector leakage assessment (TVLA) methodology in practice". In: *International Cryptographic Module Conference*. Vol. 20. 2013.

[60] Bernard L Welch. "The generalization of 'STUDENT'S'problem when several different population varlances are involved". In: *Biometrika* 34.1-2 (1947), pp. 28–35.

[61] H. Geng, J. Wu, J. Liu, M. Choi, and Y. Shi. "Utilizing random noise in cryptography: Where is the Tofu?" In: *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2012, pp. 163–167.

[62] Takashi Ishigaki, Ryuta Tsuchiya, Yusuke Morita, Nobuyuki Sugii, and Shin'ichiro Kimura. "Ultralow-power LSI Technology with Silicon on Thin Buried Oxide (SOTB) CMOSFET". In: Jan. 2010, pp. 145–156. ISBN: 978-953-307-045-2. DOI: 10.5772/54345.

[63] H. Okuhara, Y. Fujita, K. Usami, and H. Amano. "Power Optimization Methodology for Ultralow Power Microcontroller With Silicon on Thin BOX MOSFET". In: *IEEE Trans. on Very Large Scale Integration (VLSI) Syst.* 25.4 (Apr. 2017), pp. 1578–1582.

[64] SpinalHDL. *A FPGA Friendly 32-bit RISC-V CPU Implementation*. 2020. URL: https://github.com/SpinalHDL/VexRiscv.

[65] Ba-Anh Dao, Trong-Thuc Hoang, and Cong-Kha Pham. *C implementation of AES-128 on SOTB Briey SoC*. 2020. URL: https://github.com/thuchoang90/briey_software/tree/master/SOTB_Aug2019_TinyAES.

[66] *Small portable AES128/192/256 in C*. 2020. URL: `https://github.com/kokke/tiny-AES-c`.

[67] *Database of DPA contest v2*. 2010. URL: `http://www.dpacontest.org/v2/download.php`.

[68] Francois-Xavier Standaert and Tal G. Malkin and Moti Yung. "A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks (extended version)". In: *Eurocrypt 2009, Lecture Notes in Computer Science*. Cologne, Germany, Apr. 2009, pp. 443–461.

[69] D. Bellizia, S. Bongiovanni, P. Monsurrò, G. Scotti, A. Trifiletti, and F. B. Trotta. "Secure Double Rate Registers as an RTL Countermeasure Against Power Analysis Attacks". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.7 (2018), pp. 1368–1376. DOI: `10.1109/TVLSI.2018.2816914`.

[70] Kenneth Palma and Francesc Moll. "Analysis of Random Body Bias Application in FDSOI Cryptosystems as a Countermeasure to Leakage-Based Power Analysis Attacks". In: *IEEE Access* 9 (2021), pp. 114977–114988. DOI: `10.1109/ACCESS.2021.3105635`.

[71] Ckristian Duran, Hector Gomez, and Elkim Roa. "AES Sbox Acceleration Schemes for Low-Cost SoCs". In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2021, pp. 1–5. DOI: `10.1109/ISCAS51556.2021.9401539`.

[72] Zhenya Zang, Yao Liu, and Ray C. C. Cheung. "Reconfigurable RISC-V Secure Processor And SoC Integration". In: *2019 IEEE International Conference on Industrial Technology (ICIT)*. 2019, pp. 827–832. DOI: `10.1109/ICIT.2019.8755206`.

[73] Utsav Banerjee, Andrew Wright, Chiraag Juvekar, Madeleine Waller, Arvind, and Anantha P. Chandrakasan. "An Energy-Efficient Reconfigurable DTLS Cryptographic Engine for Securing Internet-of-Things Applications". In: *IEEE Journal of Solid-State Circuits* 54.8 (2019), pp. 2339–2352. DOI: `10.1109/JSSC.2019.2915203`.

[74] Xilinx. *UG1075 (v1.9) Zynq UltraScale+ Device Packaging and Pinouts, Product Specification User Guide*. Xilinx. California, United States, 2020. URL: `https:`

//www.xilinx.com/support/documentation/user_guides/ug1075-zynq-ultrascale-pkg-pinout.pdf.

[75] Ba-Anh Dao, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham. "Exploiting the Back-Gate Biasing Technique as a Countermeasure Against Power Analysis Attacks". In: *IEEE Access* 9 (2021), pp. 24768–24786. DOI: 10.1109/ACCESS.2021.3057369.

[76] Trong-Thuc Hoang, Ckristian Duran, Duc-Thinh Nguyen-Hoang, Duc-Hung Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham. "Quick Boot of Trusted Execution Environment With Hardware Accelerators". In: *IEEE Access* 8 (2020), pp. 74015–74023. DOI: 10.1109/ACCESS.2020.2987617.

[77] University of California at Berkeley. *Chipyard: An Agile RISC-V SoC Design Framework with in-order cores, out-of-order cores, accelerators, and more*. https://github.com/ucb-bar/chipyard. 2020.

[78] SiFive, Inc. *SiFive TileLink Specication*. Aug. 2019. URL: https://www.sifive.com/documentation/tilelink/tilelink-spec/.

[79] Sifive. *Freedom U540-C000 Bootloader Code*. https://github.com/sifive/freedom-u540-c000-bootloader. 2018.

[80] Joachim Strömbergson, Olof Kindgren. *Verilog implementation of the symmetric block cipher AES (NIST FIPS 197)*. https://github.com/secworks/aes. 2021.

[81] Xilinx. *7 Series FPGAs Clocking Resources User Guide UG472 (v1.14)*. July 2018. URL: https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf.

[82] Xilinx. *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics DS182 (v2.19)*. March 2021. URL: https://www.xilinx.com/support/documentation/data_sheets/ds182_Kintex_7_Data_Sheet.pdf.

[83] Jim Tatsukawa. *MMCM and PLL Dynamic Reconfiguration XAPP888 (v1.8)*. August 2019. URL: https://www.xilinx.com/support/documentation/application_notes/xapp888_7Series_DynamicRecon.pdf.

[84] Jim Tatsukawa. *Spread-Spectrum Clock Generation in Spartan-6 FPGAs XAP1065 (v1.0)*. March 2010. URL: https://www.xilinx.com/support/documentation/application_notes/xapp1065.pdf.

[85] Xilinx. *Clocking Wizard v6.0 LogiCORE IP Product Guide*. August 2021. URL: https://www.xilinx.com/support/documentation/ip_documentation/clk_wiz/v6_0/pg065-clk-wiz.pdf.

Appendix A

Related Publication

A.1 Journal

[1] **Ba-Anh Dao**, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, "Exploiting the Back-Gate Biasing Technique as a Countermeasure against Power Analysis Attacks," in *IEEE Access*, vol. 9, pp. 24768-24786, Feb. 2021.

[2] **Ba-Anh Dao**, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, "Correlation Power Analysis Attack Resisted Cryptographic RISC-V SoC with Random Dynamic Frequency Scaling Countermeasure," in *IEEE Access*, vol. 9, pp. 151993-152014, November. 2021.

A.2 Conference

[1] **Ba-Anh Dao**, Anh-Tien Le, Trong-Thuc Hoang, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, “Dynamic Frequency Scaling as a countermeasure against simple power analysis attack in RISC-V processors,” in *First International Work-shop on Secure RISC-V Architecture Design Exploration (SECRISC-V’20).* 2020.

Appendix B

Full Publication List

B.1 Journal

[1] **Ba-Anh Dao**, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, "Exploiting the Back-Gate Biasing Technique as a Countermeasure against Power Analysis Attacks," in *IEEE Access*, vol. 9, pp. 24768-24786, Feb. 2021.

[2] **Ba-Anh Dao**, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, "Correlation Power Analysis Attack Resisted Cryptographic RISC-V SoC with Random Dynamic Frequency Scaling Countermeasure," in *IEEE Access*, vol. 9, pp. 151993-152014, November. 2021.

[3] Anh-Tien Le, Trong-Thuc Hoang, **Ba-Anh Dao**, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, "A Real-Time Cache Side-Channel Attack Detection System on RISC-V Out-of-Order Processor," in *IEEE Access*, vol. 9, pp. 164597-164612, December. 2021.

B.2 Conference

[1] Binh-Nhung Tran, Ngoc-Quynh Nguyen, **Ba-Anh Dao**, and Chung-Tien Nguyen. “Implementation of XTS - GOST 28147-89 with Pipeline Structure on FPGA”. *in press*. In:*4th international conference on Modelling, Computation and Optimization inInformation Systems and Management Sciences (MCO 2021)*. 2021.

[2] **Ba-Anh Dao**, Anh-Tien Le, Trong-Thuc Hoang, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham, “Dynamic Frequency Scaling as a countermeasure againstsimple power analysis attack in RISC-V processors,” in *First International Work-shop on Secure RISC-V Architecture Design Exploration (SECRISC-V’20)*. 2020.

[3] Anh-Tien Le, **Ba-Anh Dao**, Kuniyasu Suzaki, and Cong-Kha Pham, “Experimenton Replication of Side Channel Attack via Cache of RISC-V Berkeley Out-of-Order Machine (BOOM) Implemented on FPGA”. In: *Fourth Workshop on Com-puter Architecture Research with RISC-V (CARRV 2020)*. 2020.

[4] Duc-Hoan Nguyen, Vu-Thang Nguyen, Vu Dac-Tung, and **Ba-Anh Dao**. “Novelhardware-oriented integer motion estimation algorithms for high efficiency videocoding”. In: *2019 6th NAFOSTED Conference on Information and Computer Sci-ence (NICS)*. 2019, pp. 160–165.DOI:10.1109/NICS48868.2019.9023823.

[5] **Ba-Anh Dao**, and Vu-Thang Nguyen, "A High Speed, Low Power 6-bit Flash ADCDesign”. In: *7th AUN/SEED-Net Int’l Conference on Electrical, and Electronics Engineering*. 2014.

This page intentionally left blank

Author Biography

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, Vietnam.

THE END