SECURE CRYPTOGRAPHIC DESIGNS RESILIENT TO SIDE-CHANNEL ATTACKS

by

Yutian Gui

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2021

Approved by:

_____
Dr. Fareena Saqib

_____
Dr. Kathryn Smith

_____
Dr. Tao Han

_____
Dr. Moutaz Khouja

ABSTRACT

YUTIAN GUI. Secure Cryptographic Designs Resilient to Side-channel Attacks.
(Under the direction of DR. FAREENA SAQIB)

The rapid development of IoT devices and distributed computing brings convenience and high efficiency to modern society. To enhance the security of hardware devices, quite a few cryptographic algorithms were proposed and applied. These encryption algorithms show good resilience to brute-force attacks, but are still vulnerable to side-channel attacks.

Side-channel attacks are non-invasive and passive attack that shows high efficiency on secret data extraction and brings a lot of difficulties for detection and defense. Unlike the brute-force attack and the cryptanalysis attack, that targets the weakness in the encryption algorithm, side-channel attacks utilize weaknesses of implementation and use statistical models such as differential analysis and correlation analysis to steal secret information.

In this work, we explore different side-channel attacks and propose feasible countermeasures for mitigation, including power-based analysis, electromagnetic-based analysis and Direct Memory Access(DMA) attack.

For power/EM based side channel attacks, we first demonstrate multiple attacks on both software-based implementation and hardware-based implementation, including template attack, power-based correlation analysis, and EM-based correlation analysis. To mitigate the risk, we propose a key update scheme to provide resilience to correlation-based side-channel attacks for encryption engine and prove the efficiency by experiments. To protect the process of key generation and key storage from the tampering attack, we use a secure coprocessor to generate and store secret keys.

For DMA attack, we propose a lightweight scheme to provide resilience without any physical and protocol-level modification. The proposed scheme constructs a unique

identifier for each DMA-supported PCIe device based on profiling time and builds a trusted database for authentication. The efficiency is also tested and proved by experiments.

DEDICATION

This dissertation is dedicated to all members of my family for all the support, encouragement, and many sacrifices made, especially to my:

Father: Jinxiang Gui

Mother: Min Li

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my mentor Dr.Fareena Saqib, for guiding and supporting me through my PhD career. I do appreciate her enthusiasm, extraordinary patience and serious attitude towards research, which proved to be an immense help to me, all the time.

Also, I would like to thank and acknowledge Dr.Tao Han, Dr.Kathryn Smith, and Dr.Moutaz Khouja, for being a part of my committee and taking time to give their valuable feedback.

Last but not least, I would like to express my sincere appreciation to all my friends who helped me in my research and life: Ali Shuja Siddqui, Suyash Mohan Tamore, and Geraldine Shirley Nicholas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
AMBA	Advanced Microcontroller Bus Architecture
AXI	Advanced eXtensible Interface
CEMA	Correlation ElectroMagnetic Analysis
CLK	Clock
CNN	Convolutional Neural Network
CPA	Correlation Power Analysis
CPU	Central Processing Unit
CROI	Correlation-based Region Of Interest
DCT	Decryption Counter
DDR	Double Data Rate
DEMA	Differential ElectroMagnetic Analysis
DES	Data Encryption Standard
DLL	Data Link Layer
DMA	Direct Memory Access
DOM	Difference Of Means
DPA	Differential Power Analysis
DPL	Dual-rail with Precharge phase Logic

DROI	Difference-based Region Of Interest
DUA	Device Under Attack
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECT	Encryption Counter
EM	ElectroMagnetic
EMSCA	ElectroMagnetic emanation-based Side-Channel Attack
FPGA	Field Programmable Gate Array
HD	Hamming Distance
HW	Hamming Weight
I/O	Input/Output
I2C	Inter-Integrated Circuit
ID	Identifier
IOMMU	Input-Output Memory Management Unit
IoT	Internet of Thing
ISA	Industry Standard Architecture
ISO	International Organization for Standardization
JTAG	Joint Test Action Group
LED	Light-Emitting Diode
LNT	Least Needed Traces

LNTS	Least Needed Traces for Single key
LPC	Low Pin Count
MIG	Memory Interface Generator
MLP	Multi-Layer Perceptron
MRd	Memory Read
MTD	Measurement To Disclose
NIST	National Institute of Standards and Technology
NVM	Non-Volatile Memory
OS	Operating System
PCIe	Peripheral Component Interconnect Express
PG	Power Grid
PL	Physical Layer
POI	Point Of Interest
PUF	Physical Unclonable Function
RAM	Random Access Memory
ROI	Region Of Interest
RoT	Root of Trust
RPT	Range of Profiling Time
RSA	Rivest–Shamir–Adleman
SABL	Sense Amplifier Based Logic

SCA	Side Channel Attack
SD	Shift Distance
SEMA	Simple ElectroMagnetic Analysis
SNR	Signal-to-Noise Ratio
SoC	System on Chip
SPA	Simple Power Analysis
SPI	Serial Peripheral Interface
SVM	Support Vector Machine
TA	Template Attack
TCG	Trusted Computing Group
TI	Threshold Implementation
TL	Transaction Layer
TLP	Transaction Layer Packet
TOFU	Trust-On-First-Use
TPM	Trusted Platform Module
TRNG	True Random Number Generator
UP	Update Period
VDD	Voltage Drain Drain
WDDL	Wave Dynamic Differential Logic
XOR	eXclusive OR

CHAPTER 1: INTRODUCTION

1.1 Motivation

With the rapid development of digital technologies and the increasing demand of highly integrated electronics, digitalization of traditional industries has become an irresistible trend. The widespread use of internet, computer and electronic system improves the efficiency of communication and production significantly, however highly informatization and digitalization also brings severe challenges to security and privacy, especially in areas lacking protections, such as smart home system [1] and power system [5]. A new report from McAfee shows that, the loss caused by cybercrimes is as high as $945 billion all over the world last year, and the cost of cybercrime has increased over 200% in last 8 years [6].

To enhance the security, various cryptographic techniques were proposed and have been implemented in communication and applications, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES) and RSA, in both software level and hardware level. These techniques have showed good resilience to brute-force attacks and eavesdropping attacks, as well provides high flexibility to users. Nevertheless, with the advent of more advanced attacks, such as side-channel attacks, solely relying on encryption cannot satisfy the need of privacy and security anymore.

In contrast to the brute force attack which needs millions of years to break an encryption system [7], current mainstream encryption algorithms can be compromised in a short time by collecting physical information leaked from running cryptographic devices passively and analyzing them with mathematical models [8] [9]. In addition, along with the high efficiency, recent side-channel attacks are non-invasive which makes defense and detection much more difficult than other attacks.

For mitigating the risk of side-channel attacks, many countermeasures to side-channel attacks were proposed [10] [11] [12]. However, these existing countermeasures are either too expensive or impractical in the real world [13] [14]. To address the vulnerability of side-channel attacks and enhance the security of electronic devices with low cost, this research makes a detailed study and proposes feasible countermeasures to power/EM based attacks and DMA attacks.

1.2 Contributions

Specifically, this work makes following contributions:

- Gives a detailed discussion on various side-channel attacks, and analyzes the difference among existing countermeasures for defense and detection.
- Presents attack models of different side-channel attacks, including power-based attack, electromagnetic-based attack and Direct Memory Access (DMA) attack.
- Demonstrates the vulnerability of smart IoT devices to the template attack with an experiment performed on a smart bulb.
- Performs successful correlation-based attacks on both software-based implementation and hardware-based implementation, with both real-time power consumption and electromagnetic emissions. The comparison of results and analysis is also given.
- Designs a moving target defense mechanism based on a key update scheme for mitigating the risk of correlation-based side-channel attacks. The effectiveness and efficiency of the proposed design is proved by experiments and discussed in security analysis.
- Uses a Trusted Platform Module (TPM) chip to make the key generation process more secure and provide a isolated non-volatile memory for storing keys used in the proposed key update countermeasure.

- Proposes a comprehensive scheme for mitigating DMA attacks targeting Peripheral Component Interconnect express (PCIe), including the registration process and the authentication process, without any hardware-level or protocol-level modification.

- Designs two algorithms to remove noise in collected measurements for identifier construction.

- Verifies the feasibility of the proposed countermeasure by experiments, and gives a full-detailed analysis on the overhead and limitations.

1.3 Organization

This document is organized as follows:

Chapter 2 introduces background knowledge and existing works related to topics involved in this work.

Chapter 3 compares three main categories of countermeasures to power/EM based side-channel attacks, and discusses both advantages and deficiencies of some recent works for DMA attack mitigation. The remaining chapters explain the research conducted.

Chapter 4 presents the research related to power-based and EM-based side-channel attacks, including different attack models and experiments. The vulnerability of side-channel attack explored and verified on both microcontroller-based and FPGA-based implementations. The comparison of efficiency among different attacks and platforms is also given in this section.

Chapter 5 demonstrates the proposed countermeasure to correlation-based side-channel attacks, as well as the detail of key generation and storage on the TPM chip. The effectiveness of the design is verified by experiments.

Chapter 6 describes the attack model of DMA attack shows the vulnerability by experiments. In addition, this chapter proposes a delay-based authentication scheme

to enhance the resilience of PCIe to DMA attacks.

Finally, conclusions of the research are presented in Chapter 7.

CHAPTER 2: BACKGROUND STUDIES

2.1 Side-channel Attack

In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot. The basic encryption process is to convert the sensitive information (plaintext) to a series of unreadable pattern which is called ciphertext by different algorithms. Ideally, if the security strength of encryption is strong enough, no information can be revealed without knowing the secret key used in encryption.

Traditional attacks, such as the brute-force attack, use a set of predefined values to attack a target and analyze the response until the secret information is revealed. The Data Encryption Standard (DES) was considered as a strong encryption algorithm and accepted by the government for protecting sensitive information. However, with the development of CPU technology and supercomputer, the key of DES can be easily extracted in less than 24 hours or even less [15].

To enhance the security of private data and information, new encryption standards were presented. The Advanced Encryption Standard (AES) was established in 2001 by the U.S. National Institute of Standards and Technology (NIST) and adopted as the encryption standard of the U.S. government to replace Data Encryption Standard (DES). In past years, the AES standard has been developed fully to enhance the strength of security widely applied in the communication area and data storage to protect the confidential data and provide the function of authentication. On the same computer, the attacker needs around 10^{14} times longer to extract the secret key of AES-128 than extracting the key of DES [16].

As a typical form of reverse engineering, a Side-Channel Attack (SCA) is any

attack based on leaked information (side-channel) gained from the implementation of a computing device, rather than weaknesses in the implemented algorithm itself. The concept of side-channel attack was first introduced in 1996 by Paul Kocher [17]. During execution, the leakage of physical information is inevitable, such as time delay, power consumption, electromagnetic radiation, and sound. The key concept of side-channel attacks is to find the relationship between the variation of physical parameters and hardware operation thereby steal the information. Side-channel attacks can not only exfiltrate information from the communication process but also break encryption based on analyzing the variation of parameters during the runtime.

Side-channel analysis attacks include but not limited to:

- **Cache Attack:** based on monitoring cache accesses made by the target.
- **Timing Attack:** based on measuring how much time various computations take to perform.
- **Power Analysis:** based on measuring power consumption and analyzing the relationship between hardware operations and power variation.
- **Electromagnetic Attack**based on leaked electromagnetic radiation, which can directly provide plaintexts and other information.
- **Acoustic Cryptanalysis:** based on collecting the noise made by the hardware during the running time.
- **Differential Fault Analysis:** induce faults (unexpected environmental conditions) into cryptographic implementations, to reveal their internal states.
- **Temperature Attack:** based on measuring and analyzing the variation of temperature of critical components.
- **Direct Memory Access (DMA) Attack:** utilize the feature of direct access to steal the data stored in the live memory on the victim system illegally.

The main goal of side-channel attacks is to figure out the correlated relationship between the secret-dependent leaked information and cryptographic executions on the device. To perform the side-channel attack, one prerequisite is the attacker must know the cryptographic algorithm and the implementation.

2.2 Power Analysis

Power consumption side-channel information can be used for attacking the encryption engine during computing extensive operations that produce power transients for each encryption round. As the most common side-channel, power is very easy to gain and measure for building the leakage model. The power consumption of a device reflects the aggregate activity of its individual elements, as well as the capacitance and other electrical properties of the system. By capturing power traces and modeling analysis, the hidden information can be extracted efficiently.

Figure 2.1 shows the basic attack flow of power analysis. The attacker collects the real-time power consumption of encryption and builds different mathematical models to find the relationship between the variation of power and hardware operations on the victim device thereby reveal the secret key.

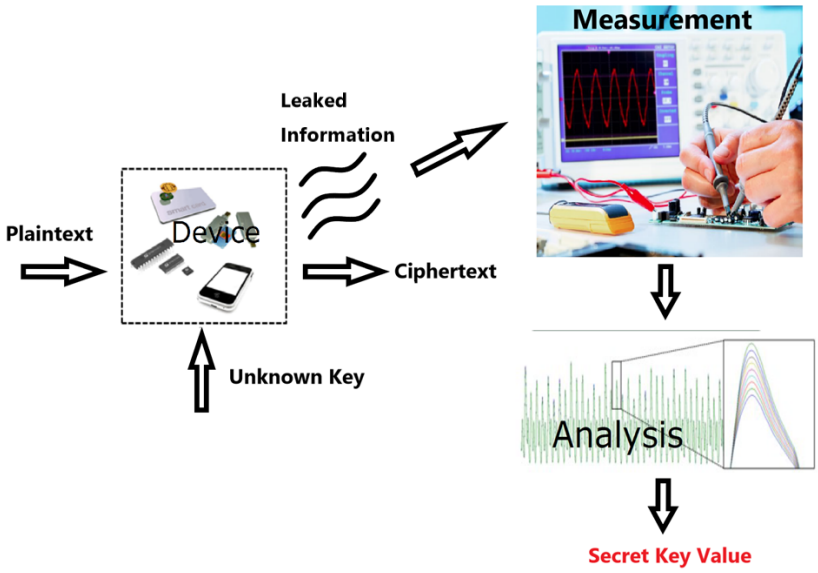

Figure 2.1: The basic attack flow of power analysis.

Typically, power-based side-channel attacks are **non-invasive**. No modifications are made to the device or system in which it is installed, and only accessible interfaces are exploited. Additionally, power analysis is a **passive** attack. The attacker only makes use of information from the power variation of the device but does not affect system resources, no evidence of the attack is presented after the attack is completed. These two features make the power analysis very difficult to detect and defend.

Currently, power-based side-channel attacks can be classified into three categories: Simple Power Analysis (SPA), Template Attack (TA), Differential Power Analysis (DPA), and Correlation Power Analysis (CPA).

2.2.1 Simple Power Analysis (SPA)

Simple power analysis is a power analysis method that extracts the secret key by looking at the variation of power consumption directly. SPA is based on the principle that different instructions and data create different patterns in the power traces that can be used for information extraction. Figure 2.2 shows an example of SPA. The attacker measures power traces from a device and extracts the key based on the physical properties of hardware devices by direct observation.

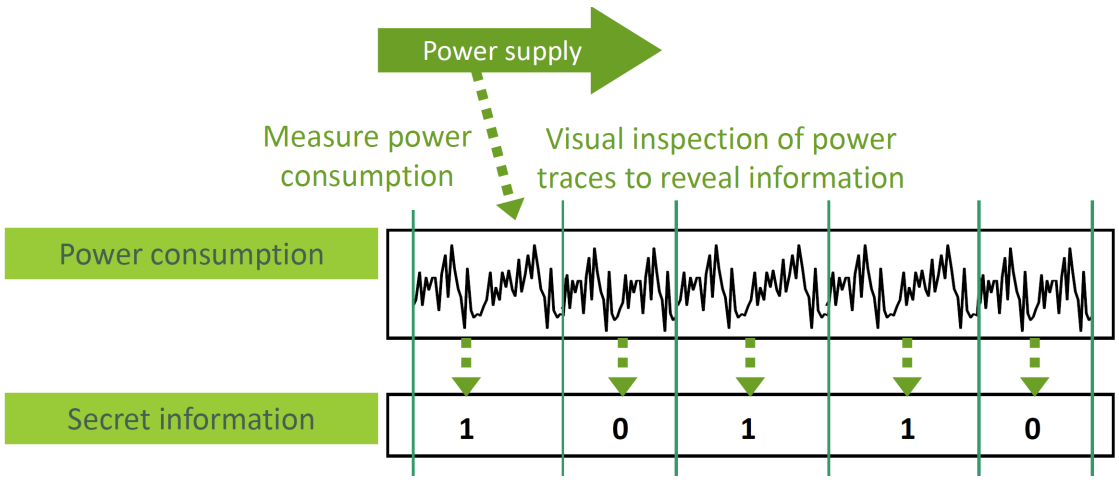

Figure 2.2: An example of simple power analysis.

SPA is only useful when the key has a significant impact on the power consumption,

and it is not inefficient when the noise is huge or on the hardware-based implementation. The software-based implementation is running in sequential processing mode, therefore the correlated relationship between the power variation and the executed operation can be captured easily. In contrast, hardware-based implementations are running in parallel and there are always multiple operation running at each moment which makes the attack more difficult [8].

2.2.2 Template Attack (TA)

As a subset of profiling attack, template attack creates a "profile" based on the leaked information (power traces, EM radiation, temperature variation, etc.) collected from the cryptographic device and applies this profile to extract the secret key of the victim quickly [18].

To perform a template attack, the attacker must have access to another copy of the protected device that they can fully control. This device is used for collecting leaked information for template (in other words, the statistical model) construction. To ensure the template contains sufficient samples, this process might take a long time but this is a one-time process. Once the template is built with enough samples, the attack only requires a very small number of new collected samples from the victim.

The efficiency of a template attack depends on the process of template construction. The key is to figure out the point or the region in each collected sample which has the highest correlation with the operation executed by the victim device (This process is also known as feature selection). This point or region is called the Point/Region Of Interest (POI/ROI).

Recently, some works [19] [20] have shown that, the efficiency of profiling attack can be improved by applying machine learning techniques in pre-processing. The machine learning algorithms, such as neural network and classification algorithms like Support Vector Machine (SVM), is able to extract the POI/ROI more quickly and efficiently which make cryptographic devices more vulnerable to template attacks.

2.2.3 Differential Power Analysis (DPA)

In addition to large scale power variations due to the instruction sequence, the variations caused by the data being executed and the instructions executed tend to be smaller and overshadowed by measurement noise. In such cases, SPA is not efficient anymore, but it is still possible to break the encryption using statistical functions.

DPA analysis uses power consumption measurements to determine whether a key block guess is correct based on differential statistical analysis and error correction techniques. Different power consumption traces (with different plaintexts but the same key) are collected for building the differential statistical model and the attacker compares each of them to remove the noise and seek the encrypted information.

Following are the stages of DPA attack:

- **Device instrumentation.**
- **Measurement:** Capture power traces with different plaintexts but the same key.
- **Signal processing:** remove alignment errors, isolate features of interest, highlight signals, and reduce noise (can be omitted).
- **Prediction and selection function generation:** apply selection functions to the cryptographic data associated with each trace.
- **Averaging:** compute the averages of the input trace subsets defined by the selection function outputs.
- **Evaluation:** analyze the DPA test results to determine the most likely candidate key guesses.

Compared to the SPA attack using primarily visual inspection to identify relevant power fluctuations, DPA shows a more efficient solution to reduce the influence caused by deceiving noise [8].

2.2.4 Correlation Power Analysis (CPA)

Correlation Power Analysis (CPA) [21] uses hamming weight or hamming distance to model the consumption of power in the device based on the assumption that the number of bits set to "0" or "1" of output is correlated with the power consumption of the device. The correlation is quantified by Pearson Correlation Coefficient between the guessed model and actual power consumption traces in sequence until the correct information is extracted. The guessed key with the highest coefficient can be considered as most likely the correct value of the secret key.

Compared to DPA, CPA has higher efficiency and it normally requires fewer power traces than DPA [21].

2.3 Electromagnetic Analysis

For the power analysis attacker is required to have physical access to the Device Under Attack (DUA). To capture the real-time power trace of the device, the prerequisite is to find an attack point or to probe the device. For example, adding a resistor between power and ground line. Figure 2.3 shows an example of power capture. To measure the power consumption, a resistor is inserted in series with the VDD pin associated with the power supply.

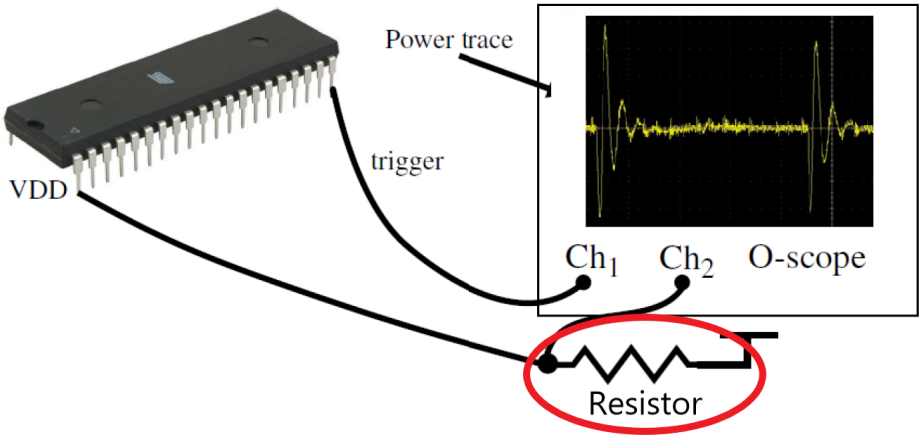

Figure 2.3: An example of power capture.

However, ElectroMagnetic emanation-based Side-Channel Attack (EMSCA) doesn't require to make any physical changes to the device. Electromagnetic radiations are captured, without touching the cryptographic device chip, by placing the EM sensor within a distance of few millimeters or even few feet in some cases. Such non-contact nature of the EM side-channel attack makes it more feasible and dangerous than power SCA as no trace is left of attack ever taking place.

EM emanations are the result of the flow of current through the different components and elements on the circuit or the microchip. Each active or passive electronic components on the circuit and data paths create their own EM radiations as well as coupled EM radiation by interfering with the nearby components EM field. This feature makes the attack based on EM radiation possible. By collecting EM radiation and performing signal analysis, the operation on the device and the information can be revealed. [22] and [23] present EM side-channel attack on a smart card chip implementing DES encryption. They applied similar techniques as used in power side-channel attack and performed Simple EM Analysis (SEMA) and Differential EM Analysis (DEMA). [24] modifies the EM attack to decrease the number of total traces required to extract key information using a pre-processing technique which reduces noise levels.

2.4 Direct Memory Access (DMA)

Before the technique of Direct Memory Access (DMA) was first introduced, computing systems were using programmed input/output and interrupt-driven I/O to communicate with peripheral devices.

For programmed input/output transmission model, all the data transfer processes are initialized and managed by the software running on the CPU. During the data transfer process, the CPU keeps monitoring and waiting until the current I/O operation is complete. This mechanism slows down the overall speed of the system, especially when the speed of CPU is much higher than the I/O component.

Different from programmed I/O, interrupts are used in interrupt-driven I/O. The I/O module sends interrupts to the CPU when new data has arrived and is ready to be read. Once the interrupt is received, the CPU will initialize the data transfer process then return to execute other tasks. Since the CPU doesn't need to wait for the completion of each transfer, the efficiency of data transmission is higher than programmed I/O. However, interrupt-driven I/O processes data in bytes, the high frequency of sending interrupts decreases the speed of CPU when the system is communicating with block devices.

Both programmed input/output or interrupt-driven I/O which require the full involvement of the CPU. The CPU reads every block of data using a peripheral bus from the I/O devices and writes it into the main memory, which degrades the performance of the system. To minimize the intervention of CPU during the data transmission, DMA DMA allows peripheral hardware devices (disk drive controllers, graphics cards, network cards, and sound cards, etc.) to send/read I/O data directly to/from main memory. In DMA model, the the CPU only takes charge of initialization and termination which frees the processor from involvement in the transfer process. The interaction between external devices and memory is carried out independently of the CPU, therefore the overload of CPU is reduced remarkably.

The feature of DMA is provided by a number of bus architectures, such as Industry Standard Architecture (ISA), Advanced Microcontroller Bus Architecture (AMBA), and Peripheral Component Interconnect (PCI). To manage the data transfer between the host system and DMA devices, a DMA controller is needed. The DMA controller is a control unit, part of the interface circuit, which enables the movement of data blocks between I/O devices and the main memory. After the initialization of the DMA controller by CPU, the memory controller provides memory addresses and initiates memory read or write cycles for data transfer, and sends an interrupt to the CPU when the whole process of data transmission is done.

2.4.1 Peripheral Component Interconnect Express (PCIe)

As the successor of Peripheral Component Interconnect (PCI), Peripheral Component Interconnect Express (PCIe) is a high-speed serial computer expansion bus standard for attaching hardware devices in a computer, such as GPUs, sound cards, hard drivers, and network interface controllers. Benefit from the higher throughput, the lower I/O pins and the better performance, PCIe is now widely used in almost all the computers.

Different from the old PCI standard which has a parallel architecture, PCIe is a serial bus standard based on differential transmission [25]. The topology structure of PCIe is shown in Figure 2.4.

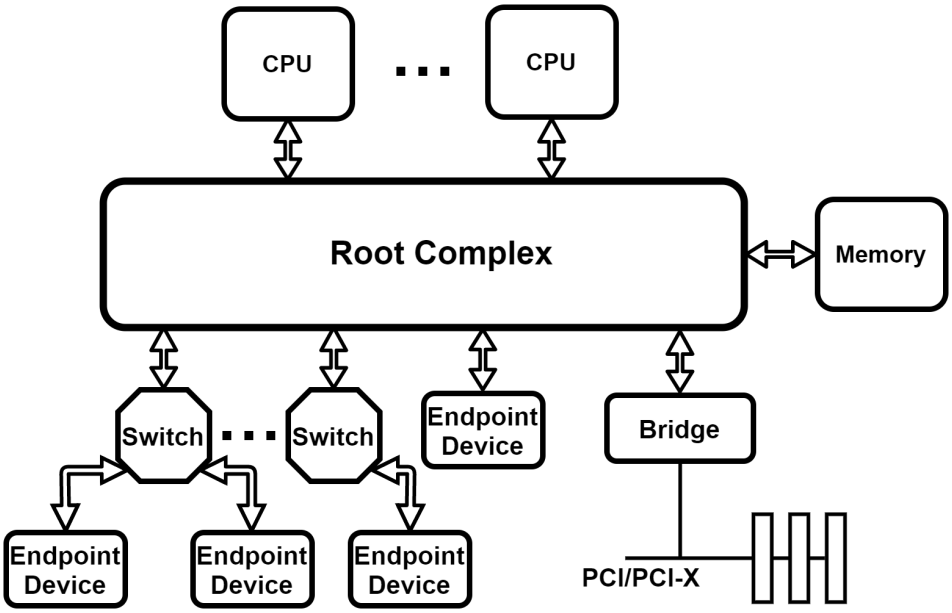

Figure 2.4: Basic PCIe topology.

The key component in PCIe is the root complex which connects the processor and the memory subsystem to PCIe switches and PCIe endpoint devices. PCIe devices, such as network adapter, graphic card and sound card, can be connected to the root complex directly, or PCIe switches. All the switches are connected to the root complex and take charge of routing incoming PCIe packets towards the correct destination [26].

PCIe also supports bridge to provide compatibility for other interfaces such as PCI, PCI-X and USB.

PCIe protocol defines three layers: Transaction Layer (TL), Data Link Layer (DLL), and Physical Layer (PL) [27]. The physical layer takes charge of encoding/decoding data, data scrambling/descrambling, serial-to-parallel conversion and parallel-to-serial conversion, etc. The data link layer is responsible for link management, error detection, flow control and power management. The transaction layer receives read and write requests from the bus and creates request packets for transmission to the link layer. Figure 2.5 shows the 3-layer structure of PCIe protocol.

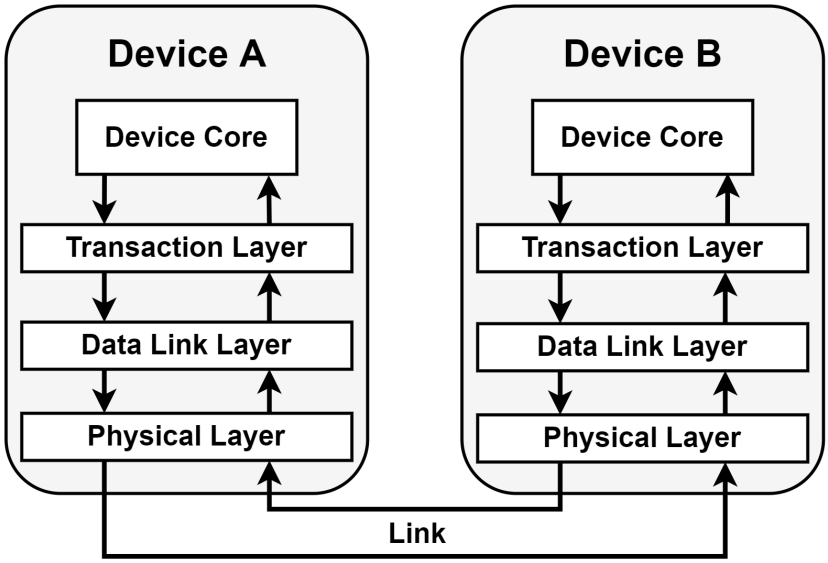

Figure 2.5: 3-layer structure of PCIe protocol.

As a bus protocol, PCI Express communication is encapsulated in packets which are defined as Transaction Layer Packets (TLPs). Figure 2.6 and Figure 2.7 show the structure of memory write request TLP and memory read request TLP. [27]

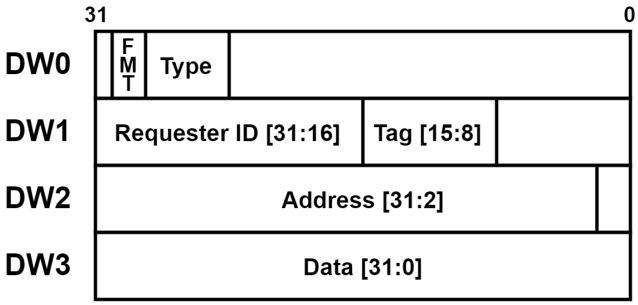

Figure 2.6: Example of memory write request TLP.

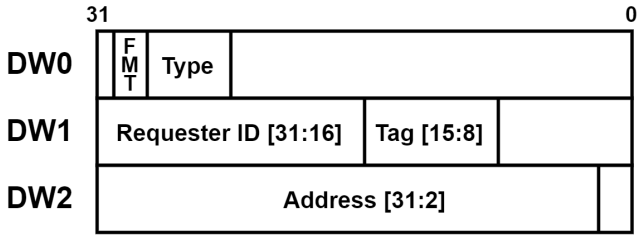

Figure 2.7: Example of memory read request TLP.

The write request TLP has 4 datawords and each one has 32 bits. The FMT field and type field define the type of this packet and Requester ID is the identifier of the sender. The Address field contains the address where the data will be written. Compared to the write request TLP, the read request TLP only has 3 datawords. The Address field has the address of memory on the receiver where the data will be read.

One the receiver side, once a new TLP is received (For example, if a PCIe device sends a memory read (MRd) request to the main memory, the main memory is the receiver, a.k.a. completer or responder), the receiver must response with the completion TLP, no matter it can or cannot fulfill the action requested. Figure 2.8 shows an example of a completion TLP.

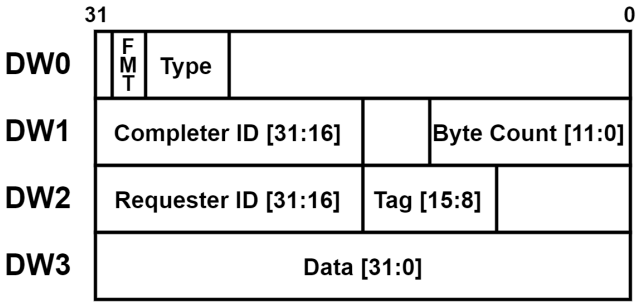

Figure 2.8: Example of completion TLP.

Each completion TLP has 4 datawords. The Completer ID is the identifier of the responder, and the Byte Count field defines the number of valid payload bytes in this packet.

2.5 DMA Attack

DMA increases the efficiency of data transfer between the main memory and external devices, but the native feature of direct access also brings some potential risks of security breaches. Due to the lack of protection, the attacker can easily gain the access to the main memory via DMA connection without the supervision from the Operating System (OS) or the CPU as well as legitimate users. As a type of side-channel attack, the DMA compromise has been proved as a powerful and efficient attack that allows the attacker to read and write the memory on the victim system directly.

In 2016, [25] demonstrated a DMA attack that allows the attacker to read/write the memory once the peripheral PCIe device is connected to the victim system without the need for hardware drivers. Moreover, this attack is able to access the live Random Access Memory (RAM) and the file system by inserting kernel implants. The same year, the Intel Advanced Threat Research team performed a DMA attack over the air by modifying a WiGig dock to compromise a laptop that is connected to the dock wirelessly [28]. The architecture of wireless connection allows the attacker to use the DMA capabilities to dump secrets out of the memory on the victim machine remotely.

The work reported in [29] shows the Thunderbolt protocol with access-control enabled is not resilient to DMA attacks. The attacker-controlled device obtains full access to the main memory successfully via a Thunderbolt port by identity clone and spoofing.

CHAPTER 3: LITERATURE REVIEW

3.1 Countermeasures to Power/EM based Side-channel Attacks

Different side-channel attacks require varying sizes of samples to achieve a successful attack that results in different amounts of time to capture and analyze leaked information. This time period is called Measurement To Disclose (MTD) period which denotes the time from the start of the physical information collection process to the end of the successful attack.

To reduce the risk of power/EM based side-channel attacks, the basic idea is to increase the value of MTD. Recently, plenty of countermeasures were proposed for mitigating side-channel attacks which can be categorized into three types: hiding, masking, and morphing.

3.1.1 Hiding

Hiding countermeasures aim to decrease the Signal-to-Noise Ratio (SNR) in order to hide information leakage in random noise [30]. In other words, the main goal of hiding is to reduce or break the correlated relationship between the processed data the dynamic leakage.

One practical method is randomization. Randomization techniques attempt to randomize the power consumption by constantly changing the execution order or by generating noise directly. In [31], the feasibility of using dynamic reconfiguration for side-channel attack mitigation was discussed. To generate random noise, a clock jitter with a dynamically reconfigurable switch matrix which determines the position of one or more registers in between functional blocks was proposed. Since each register causes a delay of one clock cycle, randomly positioning registers in between subfunctions

generate varying delays to disturb the regular clock signal. In [32], the proposed countermeasure changes the number of points in scalar multiplication in Elliptic Curve Cryptography (ECC) at each new execution of the algorithm randomly.

Another feasible way of hiding is equalization. The goal of equalization techniques is to achieve equal power consumption at each moment. [33] presents a switched capacitor circuit that equalizes the current to isolate the critical encryption activity from the external supplies, eliminating the side-channel information leakage. The result shows that the proposed design brings a significant MTD improvement of 2500x compared with unprotected circuit, but also results in additional overhead. [34] notes that the on-chip Power Grid (PG) has a vital effect on the effectiveness of power attack by inducing noise in the power profile, and presents a novel adjustment technique for PG capacitor which can regulate and control the power profile thereby reduces the power leakage. The result showed that the proposed technique increased the resilience to power-based side-channel attacks.

Dual-rail with Precharge phase Logic (DPL) [35] is another typical countermeasure based on equalization for defending side-channel attacks. Dual-rail converts the basic cell to the DR cell which conveys data by two wires for each Boolean variable. In the precharge phase, all signal wires are changed to a constant value (generally 0). In the evaluate phase, only one of the complementary signals transitions from 0 to 1, the other one remains at 0. Whatever the input and key, exactly one and only one toggle occurs. In this way, the aggregate power consumption is equalized and maintained at a relatively constant level. This method has been explored and developed for mitigating side-channel attacks by many works, such as Sense Amplifier Based Logic (SABL) [36] and Wave Dynamic Differential Logic (WDDL) [37]. Both SABL and WDDL use dual-rail and pre-charge to balance the power consumption for each gate. SABL uses a full custom logic style and all gates are connected to CLK and pre-charged altogether therefore the overhead is doubled, and the speed is also reduced.

In contrast, WDDL uses a standard CMOS library and the structure of each gate is simplified largely. AS a result, the overhead of WDDL is less than SABL but it generally less resistant against DPA attacks.

3.1.2 Masking

Countermeasures based on masking randomize the intermediate values of a cryptographic computation to avoid dependencies between these values and the leaked information. Different from hiding techniques, masking techniques are applied on the algorithmic level.

One masking technique is secret sharing, such as Boolean secret sharing and multiplicative secret sharing. The basic idea of secret sharing is to convert the original secret into a given number of masked shares using Boolean or other operations. To extract the secret, all the shares are needed because none of them alone provides enough information. [38] designed a leakage-resilient stream-cipher based on the idea of secret sharing. The proposed stream-cipher utilizes the concept of alternating extraction [39] where the size of each share is very huge, but the reconstruction phase is efficient. With the proposed design, the tolerance of leakage is highly improved.

[40] presents an order 1 perfectly masked algorithm for AES. A squaring algorithm and a multiplication algorithm are described for masking original secret data with an additively masked value in this work. The security analysis presented in this paper shows the proposed masking scheme distributes the corresponding intermediate results efficiently. However, the proposed design requires additional (fresh) random values for masking every time, it causes a high overhead for random value generation.

To increase the efficiency and the security of masking, the threshold implementation is proposed in [41] which combines the ideas of secret sharing, threshold cryptography, and multi-party computation protocols. In this work, the data is not masked by only one random value, but by two or more. The original secret data is divided into multiple shares by using the ramp scheme and the secret sharing scheme. The

original secret can be revealed only if the number of leaked shares is higher than the threshold which is determined by the scheme. This design does not need to generate fresh random values after every transformation, but the computational complexity is increased significantly, and the overhead caused by data storage is very high [42].

3.1.3 Morphing

Compared with the hardware-based cryptographic implementation, the software-based implementation is more vulnerable to side-channel attacks because of its sequential processing architecture. According to the result of some recent works, such as [2], the secret key used in AES-128 which is implemented on the microcontroller can be extracted by power analysis easily with less than 50 power traces.

To mitigate the risk of side-channel attacks at the software level, a dynamic code morphing countermeasure was introduced [43] and developed. The basic idea of code morphing is based on the technique named dynamic compilation [44] and used against power-based side-channel attacks aimed at software implementations of cryptographic engines. The core component of code morphing is the polymorphic engine which can be used to transform a program into a subsequent version that consists of different code but executes with the same functionality during the run-time. The proposed approach can increase the variability of the protected code significantly in three steps: code morphing, rescheduling, and array access permutation. However, even the Measurements To Disclose (MTD) time is increased after applying the code morphing scheme, the accompanying overhead of time delay is not negligible and results in a reduction of overall performance.

3.2 Existing Countermeasures to DMA Attacks

The vulnerability of DMA attack has been prove by experiments in some recent works [25] [28] [29]. To mitigate the risk of DMA attacks, some techniques have been proposed.

The simplest method for defending DMA attack is port locking. The basic idea of port locking is to remove all the ports for peripheral devices and interfaces. For example, one Linux, the user can disable the kernel of IEEE 1394 high speed serial bus "ohci1394", thereby blocks all the DMA access requests from peripheral devices. A more direct way is to break the integrated ports/interfaces physically. This method is absolute secure but the cost is also very huge. If all the ports are blocked or destroyed, the expansibility will be reduced significantly.

After BitLocker was first introduced in Windows Vista, Microsoft provides pre-boot authentication to enhance data privacy. The safety of data is ensured by full-disk encryption with BitLocker and the key is generated by the Trusted Platform Module (TPM) [45]. According to the policy of pre-boot authentication, BitLocker accesses and stores the encryption key in memory only after the user provides the correct PIN or USB startup key. However, BitLocker can only be used for encrypting hard drive or removable data drive, the memory is still vulnerable to DMA attacks after the booting process is completed. Moreover, the time overhead of encryption/decryption is very high.

On Windows 10, a new feature is added from version 1709 named "Disable new DMA devices when this computer is locked" [10]. Once this feature is enabled, all the hot-pluggable PCI/PCIe ports will be blocked until a user signs into Windows. The basic idea of this method is as the same as port locking, but it is more flexible and practical. The system blocks all the ports only when the Windows is in the sign-out state. This policy prevents attacks that use PCI/PCIe-based devices to access BitLocker keys, yet at the same time brings inconvenience in some cases when the user wants to keep PCI/PCIe devices working when the system is locked.

With the introduction of the Input-Output Memory Management Unit (IOMMU) (This technique is branded VT-d [11] by Intel and AMD-Vi [12] by AMD), the risk of DMA attack is reduced. As shown in Figure 3.1, IOMMU connects the DMA-capable

I/O bus to the main memory and supports DMA-remapping which translates the address of incoming DMA request to the correct physical memory address. After activating this function, each DMA device can only access a part of memory which is allocated by IOMMU therefore the rest of memory is immune to DMA attacks on that device. This technique mitigates the risk of DMA attack, but it cannot protect the whole memory especially when the memory size of the system is small. What's worse, a recent work shows that the IOMMU is still assailable to DMA attacks [46]. In this work, the attack utilizes the vulnerability in the kernel and device-driver which allows the attacker to extract private data and change the control flow by manipulating code pointers. The proposed attacks is tested on different interfaces (Thunderbolt and PCIe) and different operating systems (Linux, MacOS and Windows 10), and the result shows that the private information can be extracted on all the platforms even with IOMMU enabled.

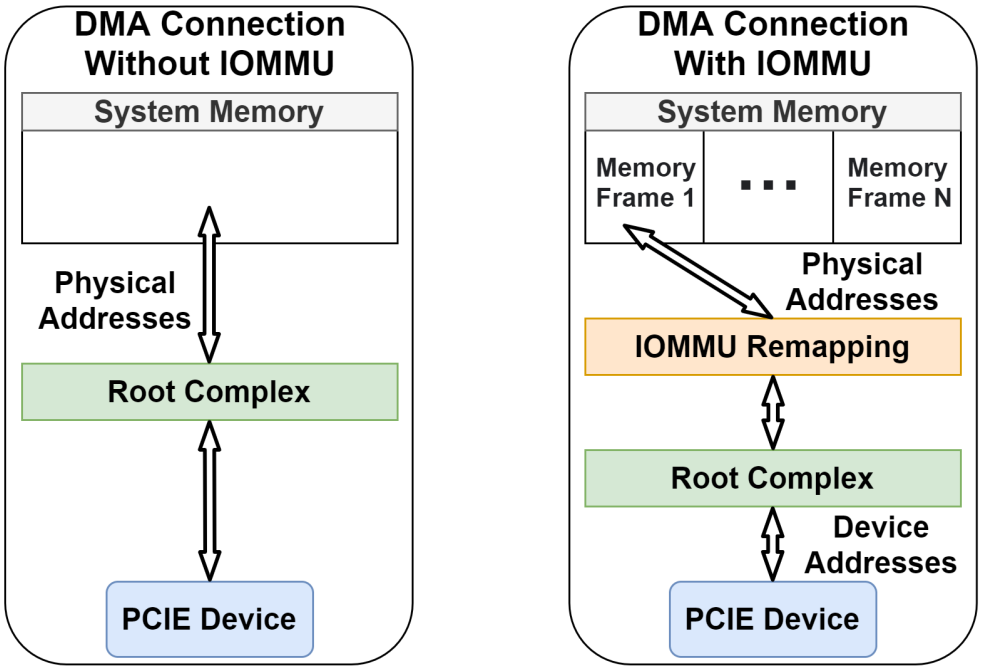

Figure 3.1: Comparison between DMA connections with IOMMU and without IOMMU.

Another strategy of DMA attack mitigation is access control which can be realized

by encryption-based authentication or Trust-On-First-Use (TOFU) scheme [47]. [13] presents an authentication architecture between each component and the host system based on the key-based certification. To ensure the security of keys, all the keys are generated with a True Random Number Generator (TRNG) on the integrated TPM. However, the precondition of this design is that each component must be authenticatable at any time which is not practical in the real world. For the TOFU scheme, the host machine records the unique fingerprint/identifier of a trusted device into its trusted database the first time this device is attached. In this way, a device can access the host system only if its identifier can be found in the trusted database otherwise it will be blocked. Nevertheless, there is no available unique identifier for devices in the case of PCIe authentication. Currently, computing systems use vendor ID, device ID, and PCIe slot ID to distinguish different PCIe devices. However, if the attacker unplugs the connected PCIe device and inserts a new device of the same model from the same manufacturer into the same slot, the host machine is not able to recognize whether the new connected device is the same as the one used previously.

CHAPTER 4: POWER/EM BASED SIDE-CHANNEL ATTACKS

In the real world, most electronic devices are unprotected to side-channel attacks, especially IoT devices and application-specific integrated circuits. As a result, the attacker can apply different attack models on victim devices to achieve different purposes maliciously.

4.1 Template Attack on Smart Devices [1]

4.1.1 Template Attack

As a powerful attack model, template attack has been applied on power traces for breaking encryption and revealing secret keys in some existing works [48] [49]. Generally, a template attack contains the following steps [18]:

1. Data collection. The power traces under different instructions and data are measured and collected. Ensure that enough traces are recorded to give information about each possible value of secret data.

2. Building the template model. The basic assumption of the template attack is that the dynamic power variation has a linear and direct relationship with the operation/data executed on the device. The message/data associated with each trace is also needed for template construction.

3. Applying template. On the victim device, record a small amount of power traces and apply the template model on each trace. For each trace, track which value is most likely to be the correct value executed on the device.

In a template attack, the most important step is the process of power trace characterization, in other words, template model construction. This is because the process

of building the template is closely associated with the overall efficiency. To extract the point or the range of points in each collected sample trace which has the highest correlation with the operations executed on the target device, different algorithms have been proposed in existing works [50]. Moreover, the efficiency of template construction can be further improved by introducing methods of feature selection which are being widely used in machine learning and image processing, such as Lasso, Elastic Net and Ridge Regression [51].

As a subset of profiling attacks, the template attack creates a "profile" of a sensitive device based on collected leaked information and extract the secret key used in cryptographic process by applying this profile [18]. In the real world, template attacks can be used for performing the simple power analysis (SPA) attack which is based on the direct observation on power traces collected during the running device.

In this work, an exploration of template attack on smart IoT devices is presented.

4.1.2 Smart Bulb

With recent advancements in smart grid and home automation, devices have become "intelligent". Generally, smart devices in the home are connected together via wireless protocols to form an IoT (Internet of Thing) network, such as Bluetooth, WiFi and Zigbee. However, for most of IoT devices, the communication between different nodes is lack of protection. As a typical representative of in-home IoT devices and endpoint devices in the smart grid, smart bulbs [52] are light bulbs that can be programmed remotely, to manipulate lighting options. Generally,

The bulb used in this work is Magic Blue UU Bluetooth Bulb which is shown in Figure 4.1.

Figure 4.1: Magic Blue Bluetooth Smart Bulb.

As shown in Figure 4.1, Magic Blue smart bulb can wirelessly connect to phone apps by using Bluetooth Low Energy (BLE). BLE is a wireless personal area network protocol aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries [53] [54]. In Bluetooth Low Energy, devices can perform one of two roles. A device can be either a "Central" (in this example, your phone) or a "Peripheral" (and respectively, the bulb) [55].

The control panel is also shown in Figure 4.1. The user can adjust the color and intensity freely by setting different values on the app. This bulb has four sets of lights: red light (R), green light (G), blue light (B) and warm-white light (W). The warm lights work alone with RGB lights, and it has no influence on the displayed color. By combing RGB lights with different intensities, a total of 16 million colors can be displayed.

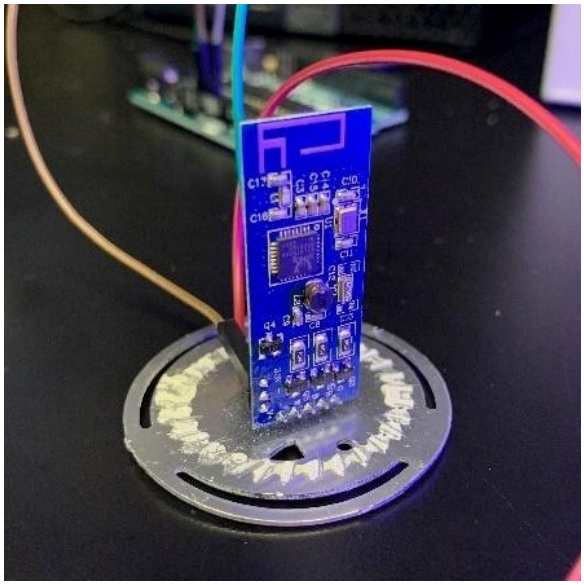

Figure 4.2: The logic board of the smart bulb.

The internal components of the smart bulb consist of the power supply, the logic board, and the LED board. The LED board consists of 3 RGB LEDs and 12 warm-white LEDs. The logic board consists of 3 pins labeled, "V+", "-", and "3.3V". Figure 4.2 shows the key component of the smart bulb which contains the logic board and the LED board. The System on Chip (SoC) used in the bulb is RTL8762AG from Realtek [56].

In the Magic UU bulb, each light has 256 levels of intensity. The color and the intensity are controlled by the co-work of RGB lights. The pins of different lights are shown in Figure 4.3.

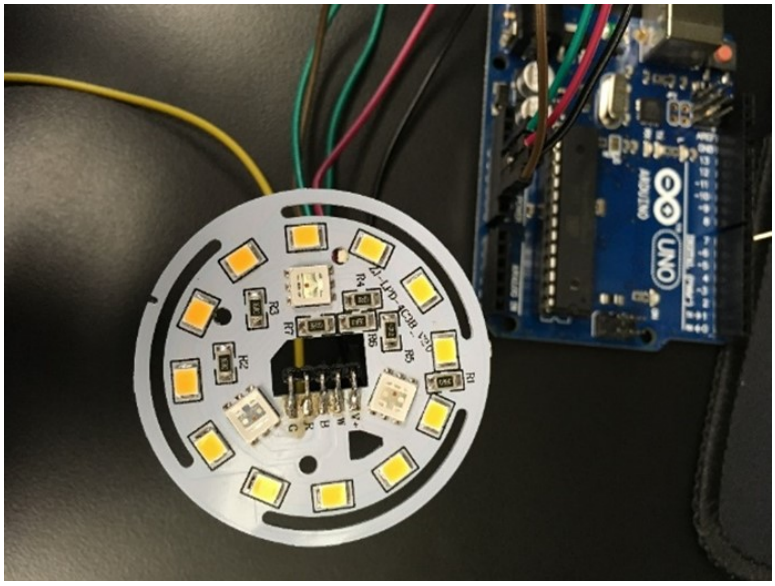

Figure 4.3: Pins of different colors: warm (W), red (R), green (G), blue (B).

4.1.3 Proposed Attack Flow

The main goal of template attack in this work is to extract the command message which controls the behave of the bulb (color and brightness). To achieve the goal, the real-time fluctuation of the voltage is used for building the mathematical model and extracting the information of command messages based on the principle that the color can be distinguished by the amplitude of the real-time voltage waveform and the intensity can be distinguished by the duty cycle.

In this work, the proposed template attack contains four steps:

1. Using a copy of the smart bulb, record a large number of real-time voltage waveforms with different command message.

2. Create a template of the device's command and build the database with all the collected waveforms correlated to each command.

3. On the victim device, capture a new waveform with an unknown command message.

4. Apply the template to the unknown trace. Compare the unknown waveform with each waveform in the database until the matched waveform is found.

4.1.4 Experimental Setup and Result

The real-time fluctuation of the voltage is collected by the oscilloscope (Figure 4.4) for building the reference model. The standard of command message is **56 RR** (red color, 256 levels of brightness) **GG** (green color) **BB** (blue color) **WW** (warm color) **0F AA**.

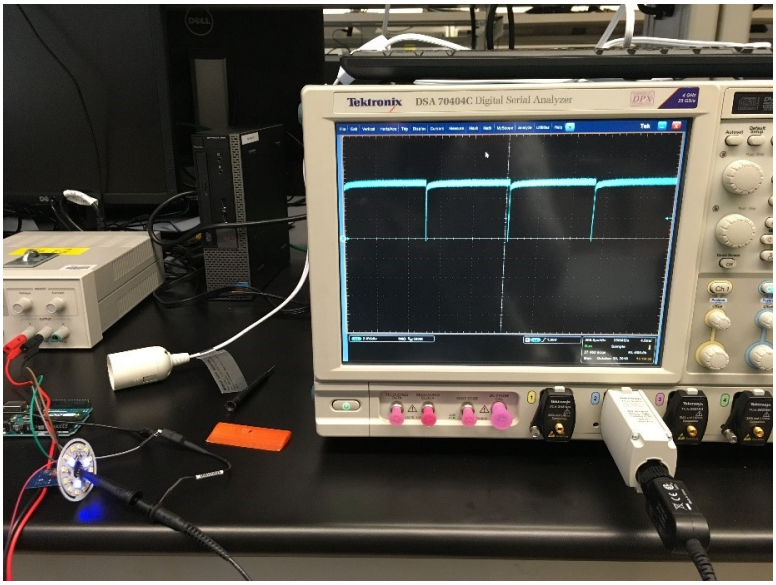

Figure 4.4: Power measurement on the Smart Bulb.

By analyzing the power trace, we found that the color can be distinguished by the amplitude and the intensity can be distinguished by the duty cycle. Figure 4.5 shows an example of waveforms with different intensities. The uppermost one is the real-time voltage waveform of red light with intensity 10, the middle one is the waveform with intensity 100 and the bottom one is the waveform with intensity 200. With the change of intensity of the light, the duty cycle also changes.

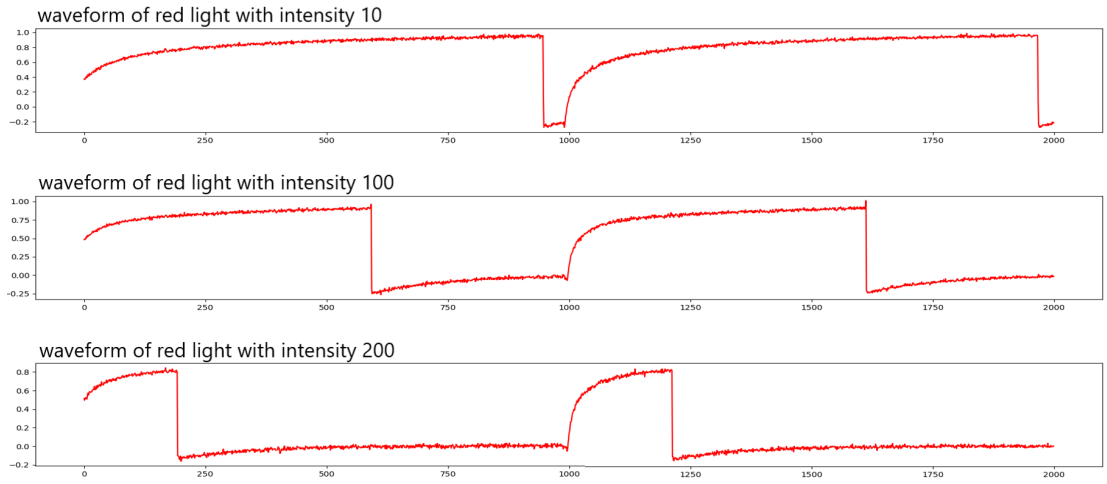

Figure 4.5: Waveforms of red light with different intensities.

In this attack, we first built the reference database of all the command messages correlated to different duty cycles and amplitudes based on the power traces we collected. Waveforms of different colors are collected and analyzed separately. The value of intensity can be deduced by comparing the duty cycle of unknown trace with the reference database and the color of the unknown trace can be deduced by analyzing its amplitude.

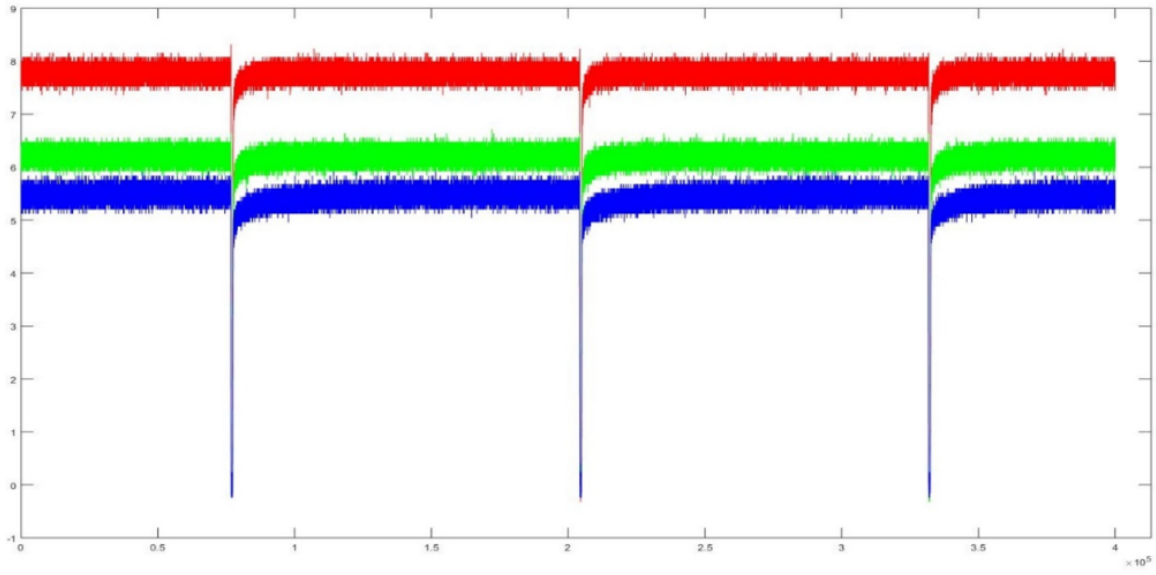

Figure 4.6: The waveform with command **56 01 01 01 00 F0 AA**.

Figure 4.6 shows another example which contains three waveforms collected from

the bulb. There are three different voltage waveforms in this figure, the red one represents the red light with 01 (Hex value) intensity, the green one represents the green light with 01 (Hex value) intensity and the blue one represents the blue light with 01 (Hex value). By comparing the amplitude and duty of these waveforms with the database we collected, we were able to extract the command sent to the bulb is **56 01 01 01 00 F0 AA**.

4.2 Template Attack on Software-based AES Implementation

Other than IoT devices, the template attack can be also applied on cryptographic implementations to reveal secret keys. In this section, a template attack on software-based AES implementation is performed.

4.2.1 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) was established in 2001 by the U.S. National Institute of Standards and Technology (NIST) and adopted as the encryption standard of the U.S. government to replace Data Encryption Standard (DES). In past years, the AES standard has been developed fully to enhance the strength of security widely applied in the communication area and data storage to protect the confidential data and provide the function of authentication. AES supports three key lengths (128, 192 and 256) to meet different requirements of security strength, using 10, 12 and 14 rounds of transformations respectively.

Figure 4.7 shows the encryption process of AES-128. The AES encryption is a streaming cipher that takes a 128-bit key (consists of 16 subkeys) and divides the data into 128-bit blocks and encrypts the data using 11 rounds as following:

1. **KeyExpansion:** Generate different round keys for each round based on the cipher key following Rijndael's key schedule.

2. **AddRoundKey:** Combine each byte of the state with the corresponding byte of the round key by using bitwise XOR.

3. **Iterative round:** Consist of 9 same iterative rounds. Each round has four operations: SubBytes (a non-linear substitution based on look-up table named S-box), ShiftRows (a cyclic shifting in each row by a certain offset), MixColumns (an invertible linear transformation combing the bytes in each column), AddRoundKey.

4. **Final round:** Similar to iterative round but only has 3 operations: SubBytes, ShiftRows, and AddRoundKey.

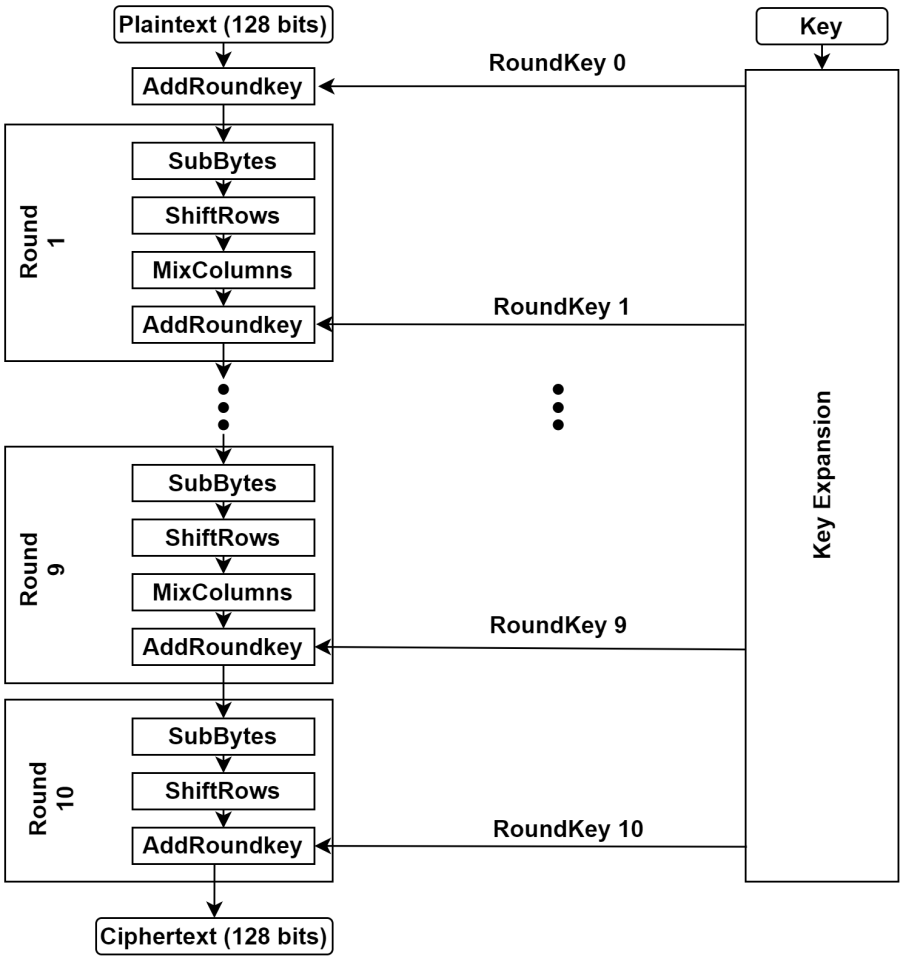

Figure 4.7: Block diagram of AES-128 encryption.

4.2.2 Proposed Attack Flow

As a type of profiling attack, power-based template attack collects the real-time power consumption of the running cryptographic device and builds the template based on all the collected power traces. The attack is divided into two main processes (training process and attack process) and following are the steps of the proposed template attack which is also shown in Figure 4.8:

1. Record an amount of power traces with **random** plaintext-key pairs. Make sure the start points and end points of all the traces are aligned.

2. Select the Point/Region Of Interest (POI/ROI) in each power traces based with the selected algorithm. Then, build the template based on the covariance matrix with the POI for all the collected power traces. In this process, the value of each random plaintext-key pair is known to the attacker.

3. On the victim device, collect a small amount of power traces (**random** plaintexts with the **same** unknown key) for attack.

4. Apply the template on power traces collected from the victim device.

5. Guess the key cumulatively until the correct key is revealed.

The quality of template depends on two factors:

- **The amount of collected power traces used in construction.** Building the template with more power traces can increase the information contained therefore improve the efficiency of attack process.

- **The algorithm used in POI selection.** The algorithm of selection decides how much valuable information can be selected for template construction. Using a proper algorithm can reduce the time overhead of attack process significantly.

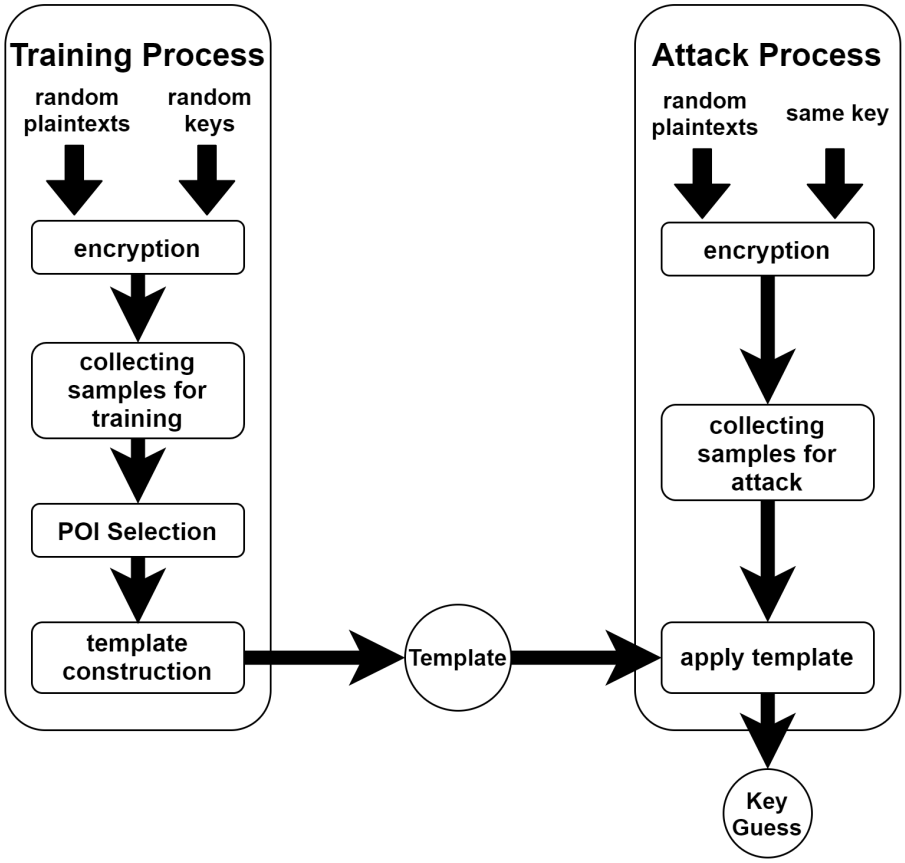

Figure 4.8: Attack Flow of the Proposed Template Attack.

To find the point which has the highest correlation with the change of the key value, Difference Of Means (DOM) based method [57] is used in this work. The algorithm of DOM is defined as follows [18]:

1. For every operation (subkeys or Hamming Weights) o and every sample point i, find the average power $P_{o,i}$. If there are T_o traces with the operation o performed, then the average power is defined as:

$$P_o, i = \frac{1}{T} \sum_{j=1}^{T_o} t_{j,i} \tag{4.1}$$

2. Calculate the sum of absolute differences S_i for each point which is defined as:

$$S_i = \sum_{o_1,o_2} |P_{o_1,i} - P_{o_2,i}| \tag{4.2}$$

3. After calculation, the point with the highest S_i value is considered as the POI.

4.2.3 Experimental Setup and Result

Encryption algorithms can be implemented at either software level or hardware level, such as Rivest-Shamir-Adleman(RSA), DES and AES. In this experiment, we implement AES-128 encryption on software-based implementation. The platform used is chipwhisperer-lite (CW1173) two-part board (Figure 4.9). The AES-128 engine is written by Python and implemented on the XMEGA 128D4 microcontroller on the target board, and the capture module is implemented on the motherboard which has a USB controller (in C) and an FPGA for high-speed captures (in Verilog) [58].

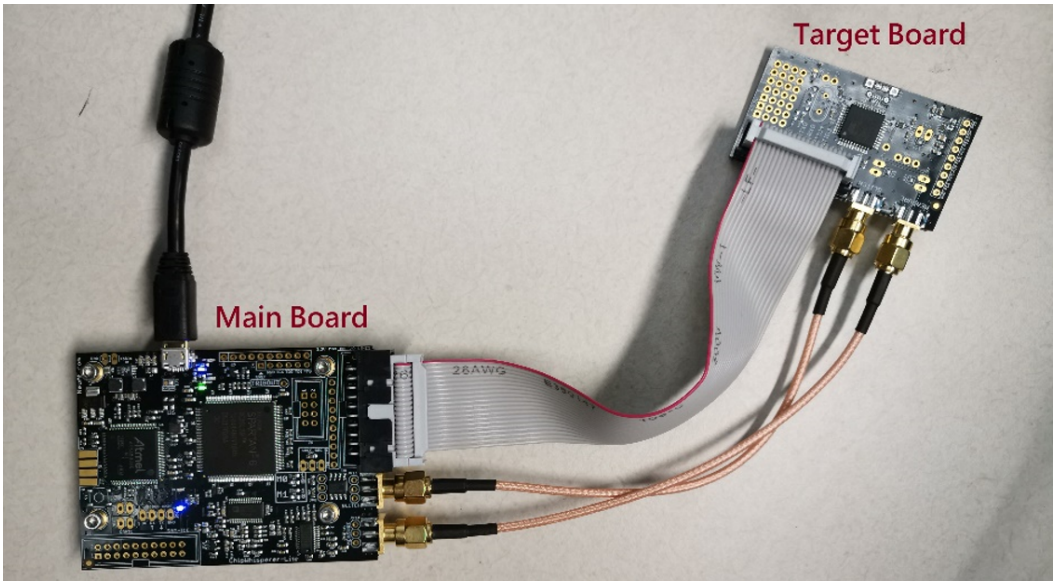

Figure 4.9: Chipwhisperer-lite (CW1173) Board.

In experiment, 2000 power traces were collected with **random** plaintext-key pairs for template construction. Each power trace contains 3000 sample points. Figure

4.10 shows a power trace of AES encryption collected from Chipwhisperer board.

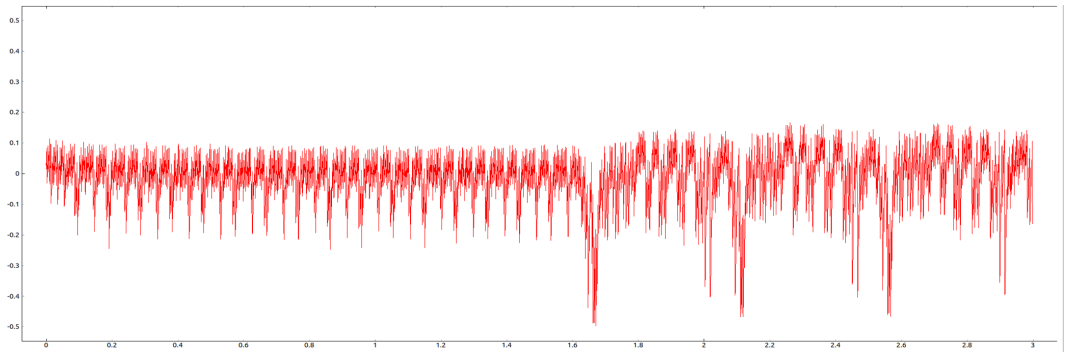

Figure 4.10: A power trace of AES Encryption on software-based implementation.

Figure 4.11 shows the result of POI selection. In this experiment, the 148^{th} point has the highest value therefore it is used for building the covariance matrix in template construction.

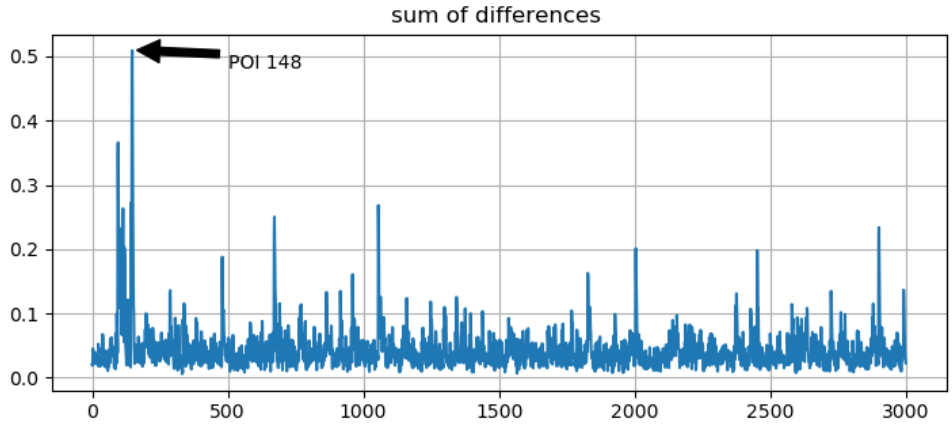

Figure 4.11: Result of POI selection.

Once the template was built successfully, another 20 traces were collected with **random** plaintexts but the **same** unknown key. Then, we applied the template on these traces one by one and guessed the correct value of the first subkey used in encryption cumulatively. The result is shown in Figure 4.12.

```
POI:  [148]
Key:  [ 21 219 144  17  52  46  50  20  70  92 194  19 202 196  21 173]
Best Match of the First Subkey (From High to Low):  [122  87  21  29  23]
Best Match of the First Subkey (From High to Low):  [153  21  29 122  87]
Best Match of the First Subkey (From High to Low):  [ 21 122  29  80 239]
Best Match of the First Subkey (From High to Low):  [ 21 122  37 239 136]
Best Match of the First Subkey (From High to Low):  [ 21 122 239  37 136]
Best Match of the First Subkey (From High to Low):  [ 21 122 239 136  93]
Best Match of the First Subkey (From High to Low):  [ 21 122 117  93  29]
Best Match of the First Subkey (From High to Low):  [ 21 122  93 239  29]
Best Match of the First Subkey (From High to Low):  [ 21 122  93 239 243]
Best Match of the First Subkey (From High to Low):  [ 21 122 243 239 117]
Best Match of the First Subkey (From High to Low):  [ 21 239 229 217  12]
Best Match of the First Subkey (From High to Low):  [ 21 114 243 217 102]
Best Match of the First Subkey (From High to Low):  [ 21 243 114 102 117]
Best Match of the First Subkey (From High to Low):  [ 21 114 117 245 122]
Best Match of the First Subkey (From High to Low):  [ 21 114 117 122 102]
Best Match of the First Subkey (From High to Low):  [ 21 114 217 102 122]
Best Match of the First Subkey (From High to Low):  [ 21 114   1 217 117]
Best Match of the First Subkey (From High to Low):  [ 21 217 114 117 102]
Best Match of the First Subkey (From High to Low):  [ 21   7  31 117 217]
Best Match of the First Subkey (From High to Low):  [ 21   7   8 114 102]
```

Figure 4.12: Result of the proposed template attack on the first subkey.

The result shows that, after applying the template, the value of the first subkey (**21**) can be extracted correctly with only 3 power traces.

4.3 Correlation Power Analysis (CPA) on Software-based AES Implementation [2]

4.3.1 Proposed Attack Model

Benefit from using Pearson Correlation Coefficient in model building, the correlation-based side-channel attack has a very high efficiency on key extraction. Following are steps of CPA attack:

1. The power traces are normalized using pre-amplifier and collected by oscilloscope during the execution of the processing encryption.

2. Make the key prediction. The original key is divided into 16 subkeys. For each subkey, guess every possible value.

3. Predict the power consumption using the Hamming Weight (HW) leakage model to extract dynamic power consumption which reflects the data moving and operation. Hamming weight model presented in [59] states the correlation between

data processed by the CMOS device and the electricity consumed at the same time. The mathematical equation for the hamming weight model is defined as:

$$E = aH(x) + b \tag{4.3}$$

where E is the electricity consumption and $H(x)$ is the Hamming weight of the data. One important thing to note is, the leakage of power can be also modeled with other metrics, such as Hamming Distance (HD) and Shift Distance (SD) [60].

4. Evaluate the correlation between the modeled power and the actual power trace by using the Pearson correlation coefficient ρ which is defined as:

$$\begin{aligned} \rho(A,B) &= \frac{cov(A,B))}{\sigma_A \sigma_B} \\ &= \frac{E[(A-\mu_A)(B-\mu_B)]}{\sqrt{E[(A-\mu_A)^2]}\sqrt{E[(B-\mu_B)^2]}} \end{aligned} \tag{4.4}$$

where A and B are variables, cov denotes the covariance, σ denotes the standard deviation, μ is the mean value and E is the expectation value. In CPA attack, two variables used in calculation are the hypothetical value and the actual power trace, so the Pearson correlation coefficient is applied in this way:

$$C(h,t) = \frac{\sum_{d=1}^{D}[(h_d - \bar{h})(t_d - \bar{t})]}{\sqrt{\sum_{d=1}^{D}(h_d - \bar{h})^2 \sum_{d=1}^{D}(t_d - \bar{t})^2}} \tag{4.5}$$

where h is the hypothetical value of the subkey, t is the power trace, D is the total number of collected power traces.

In CPA attack, the guessed subkey with the highest coefficient is considered as most likely the correct subkey used in encryption. Overall, the accuracy of CPA attack is associated with two factors:

- The amount of collected power traces used in the attack model.
- The difference of lithography technology. The chips manufactured with advanced lithography technology (for example, 14 nm and 28 nm) leaks less physical information than chips manufactured with old lithography technology (for example, 45 nm and 65 nm) [61].

In this work, the CPA attack is applied on real-time power traces collected from software-based implementations of AES-128.

The correlation-based side-channel attack can be performed on the first round (with known plaintexts) or the last round (with known ciphertexts) in AES encryption. In this work, all the correlation-based attacks (including power analysis and electromagnetic analysis) are performed on the first round with known plaintexts. In the encryption process of AES, most of the energy is consumed by the operation of SubBytes [62], so the output of the S-Box is what we will use to check the guessed value of the key which is shown in Figure 4.13.

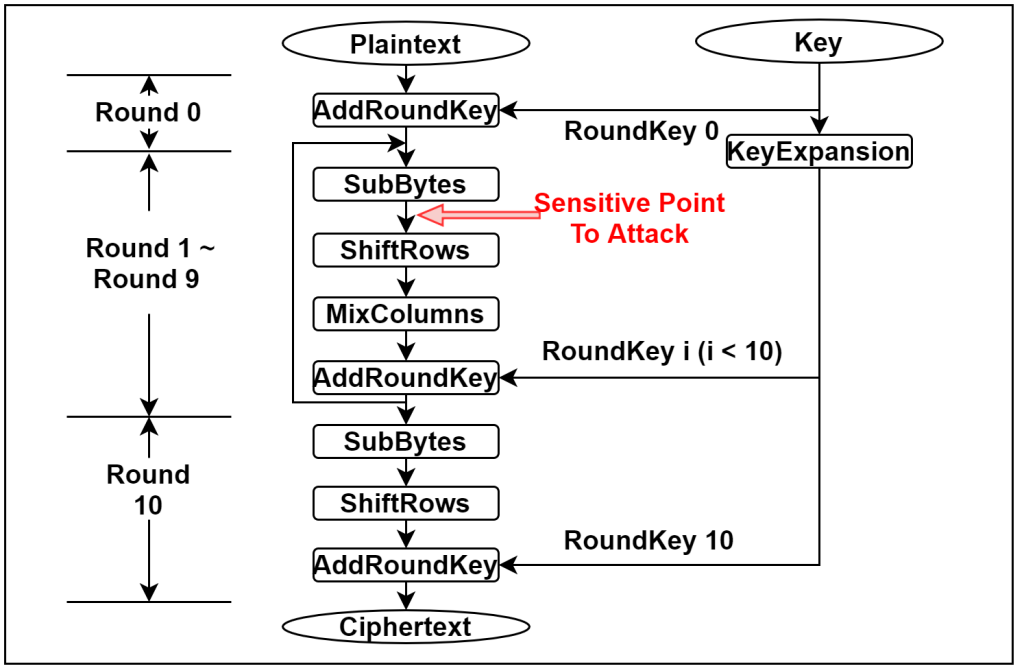

Figure 4.13: The attack point in proposed attack model.

4.3.2 Experimental Setup and Result

In this experiment, the experimental setup is the same as template attack presented in Section 4.23. Different from the template attack, the CPA attack is an accumulative process. The collected power traces are applied to the mathematical model one by one.

In CPA attack, the original key is divided into 16 subkeys and attacked separately. The result of CPA attack with different numbers of collected power traces on software-based implementation is shown in Figure 4.14.The x-axis represents the accumulated quantity of power traces used in CPA attack, and the y-axis represents the number of revealed subkeys. With 10 power traces, none of subkey is revealed correctly. However, after applying 40 power traces in the CPA attack model, all the subkeys are extracted correctly.

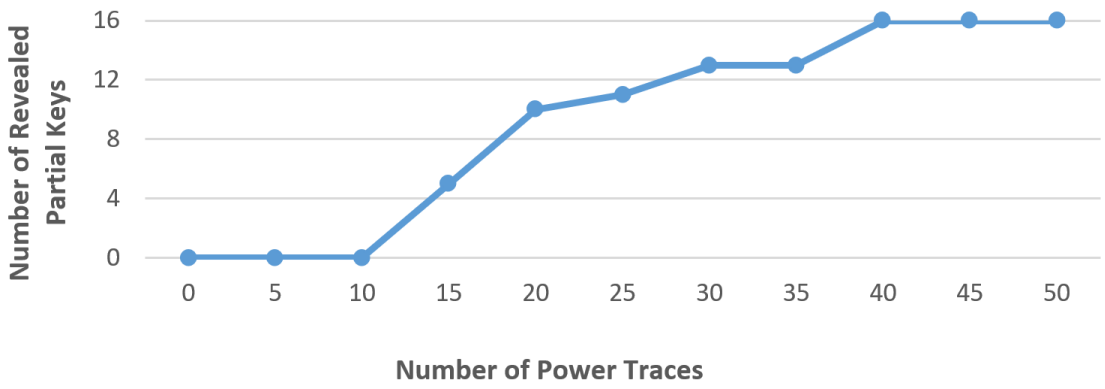

Figure 4.14: The Result of the proposed CPA attack on software-based AES implementation.

According to the experimental result, the key used in software-based AES-128 implementation can be extracted correctly by CPA attack with less than 40 power traces. It means that implementing AES encryption at the software level is very vulnerable to side-channel attacks.

4.4 Correlation Power Analysis (CPA) on Hardware-based AES Implementation

The experimental result shows that the software-based implementation has low resilience to side-channel attacks. In this work, the tolerance to side-channel attack on the hardware-based implementation is also explored. In this experiment, the AES engine is implemented on the Kintex-7 160T FPGA chip based on 28nm technology (in Verilog), the controller and the trigger is implemented on the Spartan-6 FPGA chip on the Sakura-X experimental board [63]. The power consumption is amplified by the LNA-1050 low noise amplifier [64] and captured by the DSA 70404C oscilloscope [65] at a sampling rate of 6.25 GS/s. Figure 4.15 shows the detail of the setup for capturing power consumption.

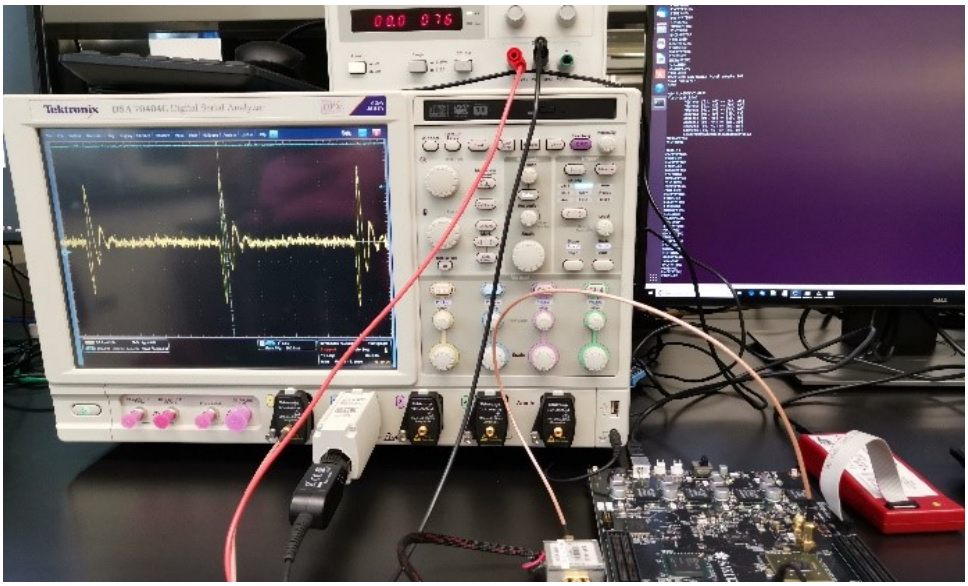

Figure 4.15: Experimental setup of power capture on FPGA-based implementation.

In this experiment, we collected 30000 power traces with random plaintexts (known to the attacker) but the same unknown secret key (2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C) considering the better tolerance of hardware-based implementation. Figure 4.16 shows an example of a power trace of AES-128 encryption process.

Figure 4.16: Power trace of AES-128 encryption implemented on Kintex-7.

After applying the CPA attack on the collected power traces, the result is shown in Figure 4.17. The value of the first subkey used in encryption is 2B (43 in decimal), and it takes around 5000 power traces to break it correctly. The graph shows that as we include more traces, the correlation coefficient of the correct key combination is much higher than other combinations which reach close to 0 coefficient.

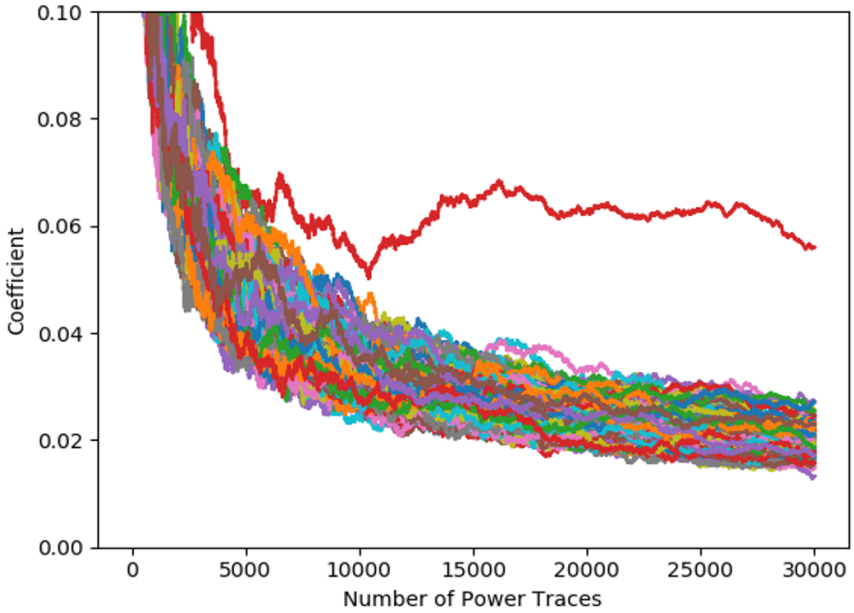

Figure 4.17: The result of the CPA attack on the first subkey of the key used in the AES-128 encryption.

As shown in Figure 4.17, the required number of power traces on FPGA-based AES implementation (around 5000 traces in this case) is much more than the attack on the software-based implementation (around 50 traces). However, even the hardware-based implementation shows better resilience to the CPA attack, it is still unsecure.

4.5 Electromagnetic Analysis on Hardware-based AES Implementation

To verify the effectiveness of EMA on the FPGA-based implementation of AES-128, the CEMA is also explored. EM emissions are the direct cause of the energy consumption of the CMOS device and as described in [23], a correlation exists between EM signal peaks and the data under process. Hence, we can consider EM emission instead of power consumption in the equation given in [59] and use hamming weight to build EM leakage model. The basic model of CEMA attack is similar to the CPA attack with the following differences:

- Different from the passive probe used in CPA attack, the CEMA attack uses a specialized EM probe.
- The process of EM capture is **non-contact** and the EM radiation is more susceptible to the environmental noise, so the amplification factor in EM capture is higher than power collection.
- In the process of EM collection, the EM probe captures all the radiation from all the components even some of them are not related to the encryption process. Therefore, an additional pre-processing is needed to remove/reduce the noise generated by other component.

To explore the difference and the efficiency of EM-based side-channel attacks, we performed Correlation ElectroMagnetic Analysis (CEMA) on hardware-based implementation of AES-128. To capture EM traces, we used CW505 Planar H-field probe [66] manufactured by NewAE technologies. The setup of EM capture is shown in Figure 4.18.

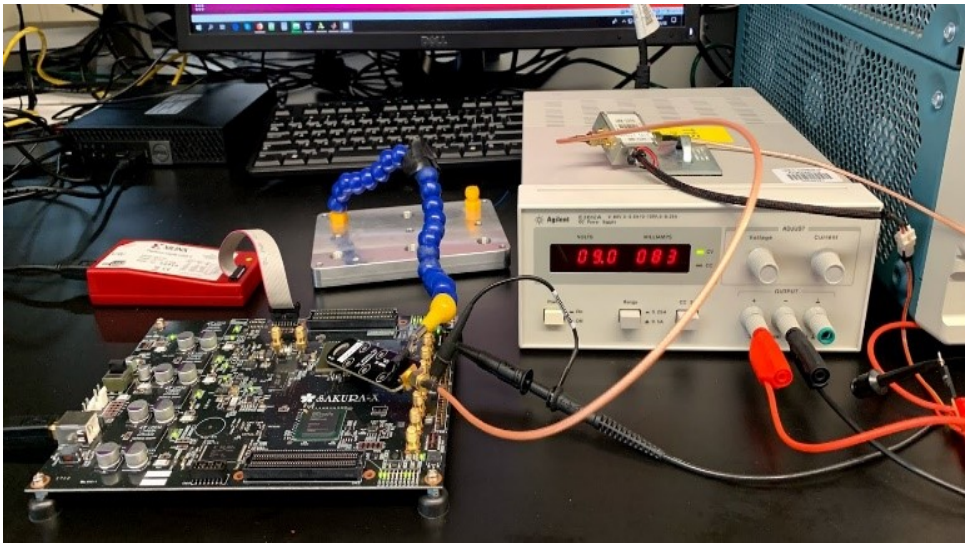

Figure 4.18: Experimental setup of EM capture on FPGA-based AES-128 implementation.

30000 EM traces were collected in this work. To compare the efficiency of CEMA attack with the CPA attack, the EM traces collected for the CEMA attack used the same set of plaintexts and the same key used in the CPA attack (2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C).

Figure 4.19 shows an example of a EM trace of the whole AES-128 encryption process. The blue line is the trigger signal, and the yellow line is the EM radiation of the chip. As discussed in Section 2.3, the EM probe captures all the EM radiation from all the components on the target device, no matter it is related to the encryption operations be executed. In this experiment, the EM radiation caused by the trigger signal is also captured (the first and the second peak of yellow waveform in Figure 4.19), so the first two peaks are removed and the attack begins from the third clock cycle.

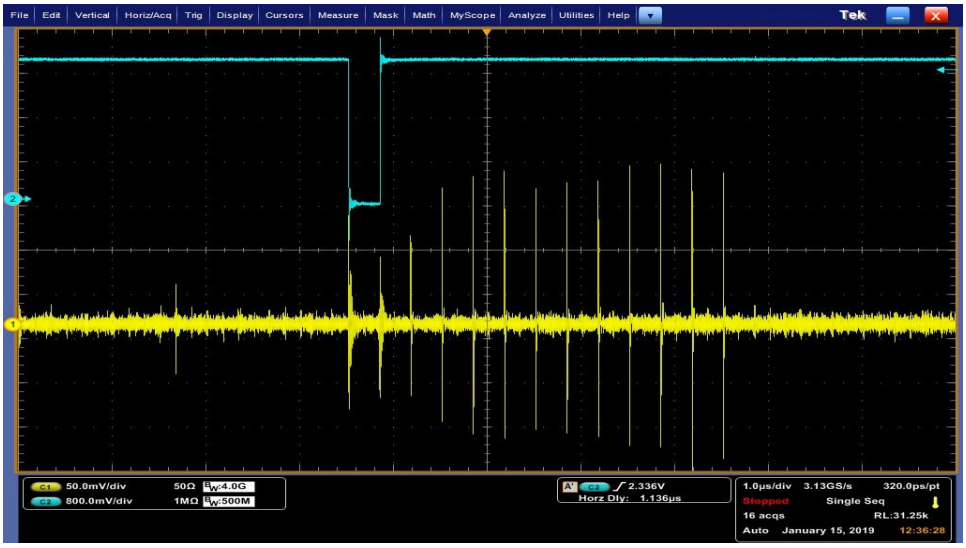

Figure 4.19: EM trace of AES-128 encryption implemented on Kintex-7.

Then we applied CEMA on the first subkey of the AES key with the collected EM traces to explore the efficiency of the EM-based side-channel attack. The result is shown in Figure 4.20.

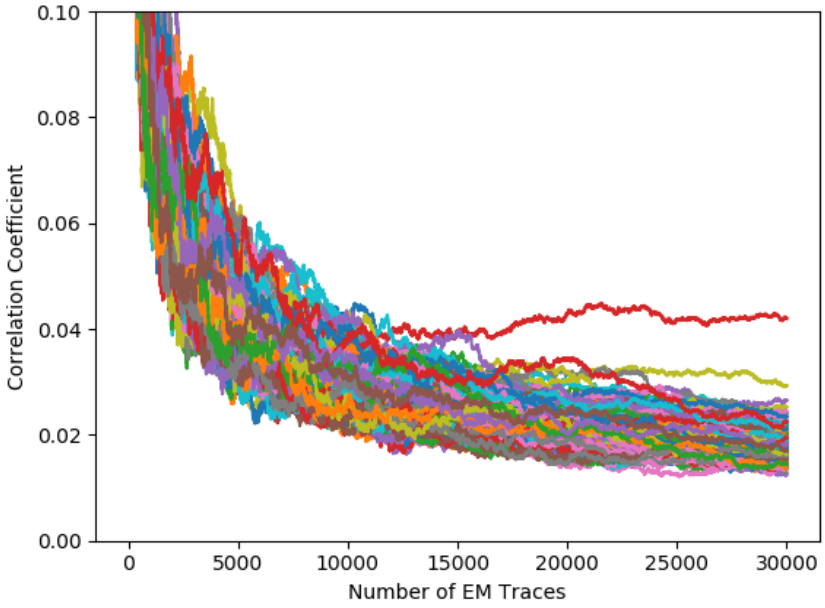

Figure 4.20: The result of the CEMA attack on the first subkey of the key used in the AES-128 encryption.

For the same key, the required number of traces for a successful CEMA attack is

around 15000 which is much higher than the CPA attack (around 5000 power traces).

4.6 Security Analysis of Power/EM Based Side-channel Attacks

In this section, the performance and vulnerability of different power/EM based side-channel attacks are compared and analyzed in detail. Table 4.1 gives the comparison of experimental results among different attacks.

Table 4.1: Comparison between different power/EM based attacks

Attacks	Target	Key Value	Least Needed Traces for a Successful Attack
CPA Attack on Software-based AES-128	XMEGA Microcontroller	2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	< 40
Template Attack on Hardware-based AES-128	Virtex-5 FPGA chip	Training: Random Keys Attack: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	Training: 500000 Attack: Failed
CPA Attack on Hardware-based AES-128	Kintex-7 FPGA chip	2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	≈ 6000
Template Attack on Software-based AES-128	XMEGA Microcontroller	Training: Random Keys Attack: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	Training: 2000 Attack: 3
CEMA Attack on Hardware-based AES-128	Kintex-7 FPGA chip	2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C	≈ 15000

As shown in Table 4.1, the CPA attack shows high efficiency on software-based encryption algorithms. With known plaintexts, the proposed CPA attack needs around 40 power traces to extract all the subkeys correctly. As a comparison, on the same device and implementation, the proposed template attack only needs 3 power traces to reveal the correct value of the key used in the encryption which is less than CPA attack.

However, since the efficiency of template attack is based on the template constructed in the training process, so the time consumed in this process is also considered as time overhead. In this case, we used 2000 power traces with random plaintext-key pairs to construct the template and 20 traces with random plaintexts but the same key in attack process. Figure 4.21 shows the time overhead of training process and attack process.

```
Template is ready!
Elaspsed Time on Template Construction:  0.054816246032714844
Key:  [ 21 219 144  17  52  46  50  20  70  92 194  19 202 196  21 173]
Best Match of the First Subkey (From High to Low):  [122  87  21  29  23]
Cumulaive Elapsed Time on Attack 0.03993487358093262
Best Match of the First Subkey (From High to Low):  [153  21  29 122  87]
Cumulaive Elapsed Time on Attack 0.09873580932617188
Best Match of the First Subkey (From High to Low):  [ 21 122  29  80 239]
Cumulaive Elapsed Time on Attack 0.15863394737243652
Best Match of the First Subkey (From High to Low):  [ 21 122  37 239 136]
Cumulaive Elapsed Time on Attack 0.21643638610839844
Best Match of the First Subkey (From High to Low):  [ 21 122 239  37 136]
Cumulaive Elapsed Time on Attack 0.2732841968536377
```

Figure 4.21: Time overhead of template attack.

In this experiment, the time consumed in training process is around 0.055 seconds and the attack process used around 0.159 seconds to reveal the subkey correctly. It is clear that the time consumed in training process is not high. Additionally, considering template construction is a one-time process, the time overhead of training is totally acceptable.

Compared to software-based implementation, the CPA attack on hardware-based implementation requires more power traces. From Table 4.1 we can know that, the same CPA attack needs around 6000 power traces to extract the same secret key used

in a FPGA-based AES-128 implementation which is much more than software-based implementation.

We also tested the same template attack on power traces collected from the hardware-based implementation. The power traces used in this experiment were collected from a SASEBO-GII board (the AES is implemented on a Virtex-5 chip) [67] and provided by DPA contest V2 [68]. With 500000 power traces with random plaintext-key pairs used in template construction, we were still not able to extract the correct value of any subkey with up to 10000 traces with random plaintexts but the same key in attack process.

The main reason is that the software-based implementation works **sequentially**, the real-time power trace leaks more information which makes the process of finding the statistical relationship between the measured signal and the hypothetical model faster and easier. By contrast, FPGA works in **parallel** therefore the real-time power consumption reflects the aggregate activity of all the executing operations at each at each moment which makes it more resilient to side-channel attacks.

Different from power-based side-channel attack, CEMA uses electromagnetic radiation collected from the running cryptographic device for attack. In this work, the feasibility and the efficiency of CEMA are tested by experiment. As shown in Section 4.5 and Table 4.1, the CEMA needs around 15000 EM traces to extract the secret key. This number is much higher than CPA attack which only needs around 6000 power traces.

There are two reasons to explain this difference:

1. The EM probe used in this experiment (CW505 Planar H-Field Probe) doesn't have enough sensitivity.

2. The level of environmental noise is too high. Different from power-based analysis, the EM probe collects EM radiation from all the components nearby no matter they belong to the target device or not.

However, if the environmental noise can be isolated and the probe is sensitive enough, the CEMA attack can reach the same level of efficiency as CPA. [61] To realize this, the EM probe must be sensitive enough and the whole device needs to be placed in a full-isolated environment.

Compared to power-based side-channel attacks which always require a physical contact point for power measurement, the measurement process of EM is **non-contact** which makes the detection of CEMA attack more difficult.

CHAPTER 5: COUNTERMEASURE TO CORRELATION-BASED SIDE-CHANNEL ATTACKS [3]

To mitigate the risk of correlation-based side-channel attacks, we propose a key update scheme on the hardware-based implementation of AES encryption. The proposed design is based on the idea of target moving and uses a secure co-processor to protect the key generation process and provides an isolated memory for key storage.

5.1 Key Update Scheme

Different side-channel attacks require varying sizes of sample traces to achieve a successful attack that results in different amounts of time to capture and analyze leaked information. This time period is called Measurement To Disclose (MTD) period which denotes the time from the start of the physical information collection process to the end of the successful attack.

We use the Least Needed power/electromagnetic Traces (LNT) to quantify the least amount of time needed for a successful attack. In other words, the less the amount of collected traces used for revealing the key, the higher efficiency the performed attack has.

To mitigate the risk of side-channel attacks, the main target in this work is to increase the LNT. We propose a key update scheme to achieve this goal. This scheme is applied and implemented on both the sender side and the receiver side, changes the value of the key used in AES encryption/decryption before the last used key can be revealed by the side-channel attack and also preserve forward secrecy.

Following are the steps of the proposed scheme which is also shown in Figure 5.1:

1. Determine the LNT for Single key (LNTS) of the target hardware. This process

is repeated several times and the average value will be used as LNTS.

2. Generate a list of random secret keys for encryption/decryption on TPM, and share the key list with the receiver.

3. Set the Update Period (UP) which is less than LNTS and share it with the receiver.

4. Start the encryption process with the first key and change the key following the order of the key list when the value of the counter reaches the value of UP on both the sender side and the receiver side. The counter ECT and DCT are used for recording the number of encryption processes have been completed with the current key on the sender side and the receiver side, respectively.

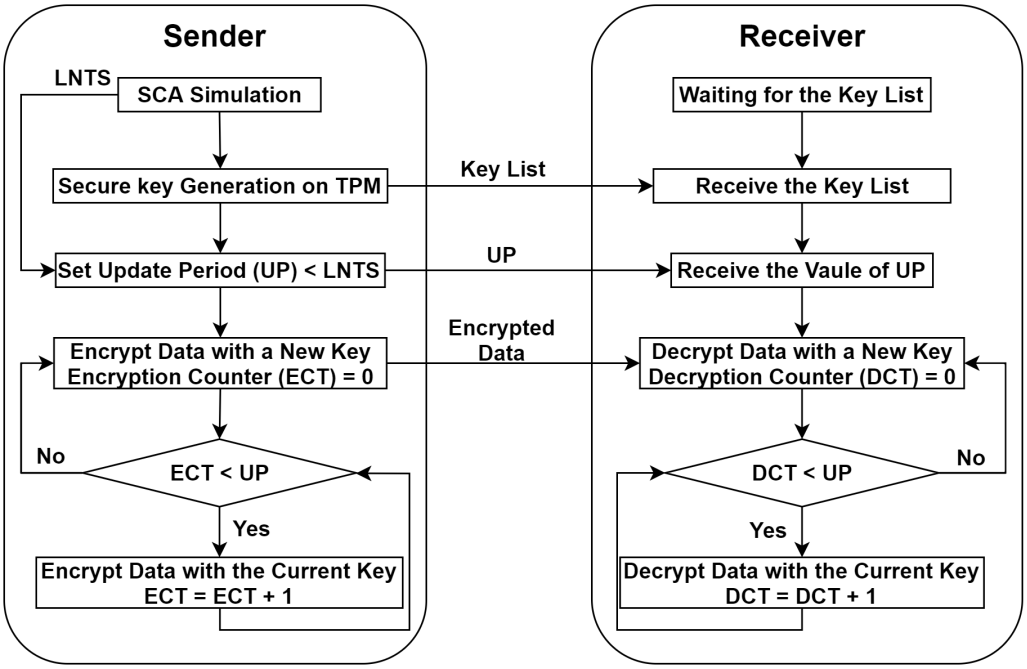

Figure 5.1: Proposed key update scheme.

Moreover, to protect keys used in key update, the proposed design uses a Trusted Platform Module (TPM) to generate and store keys. The detail of TPM and experimental setup are discussed in Section 5.2. By applying this scheme, the keys are

generated randomly and stored in a full-protected environment and the attacker has no enough time to break any key before the current key is replaced by a new random key.

5.2 Trusted Platform Module (TPM)

Trusted platform module (TPM) is a security co-processor chip specified by the Trusted Computing Group (TCG) [69] and standardized by the International Organization for Standardization (ISO) for enhancing the security of hardware devices. TPM contains a built-in Ture Random Number Generator (TRNG), a tamper-resistant Non-Volatile Memory (NVM) and a serious of functionalities to realize Root of Trust (RoT).

TPM supports various encryption standards to meet security requirements, such as AES, RSA encryption and hash function to provide Root of Trust (RoT) [70] and authentication for hardware devices and communication [71]. Currently, the TPM standard has been moved to version 2.0. Compared with TPM 1.2, TPM 2.0 enables greater crypto-agility by supporting more and newer cryptographic algorithms. Beyond that, TPM 2.0 has a three-level hierarchy architecture and allows multiple keys and algorithms per hierarchy. Keys used for encryption and authentication are derived from the primary keys and can the in the tamper-resistant persistent memory on the TPM. For generating Elliptic Curve Diffie Hellman (ECDH) session keys, TPM can use NISTP256 and BNP256 curves to generate public-private key pairs. The public key of the communicating node is multiplied with a node's own private key to generate a symmetric AES key.

The key update scheme requires multiple keys, so the security of key generation and key storage is critical. The keys can be generated on the fly using embedded structures such as a strong PUF [72] or can be stored in non-volatile memory or the secure memory on processor. The memory on processor is vulnerable to readout using the test structures such as scan-chain [73] and JTAG [74] used for testing the

hardware. Even with the secure design components, the keys may be vulnerable to side-channel analysis techniques that reveal key location or behavior during execution [75].

In this work, we integrate the TPM chip with the FPGA fabric which provides secure key generation and storage for the key update scheme. TPM supports encryption and authentication, also has a tamper-resistant non-volatile memory for key storage. The integration can be done at different levels of abstraction that is hardware bare metal or supported to core operating system functions. We demonstrate the integration on the Microblaze based system, consisting of a programmable logic based Serial Peripheral Interface (SPI) core that is connected to the Microblaze with the help of the AXI interconnect. A Memory Interface Generator (MIG) is used to connect the on-board DDR RAM. Pins from the SPI core are connected to the TPM 's SPI interface.

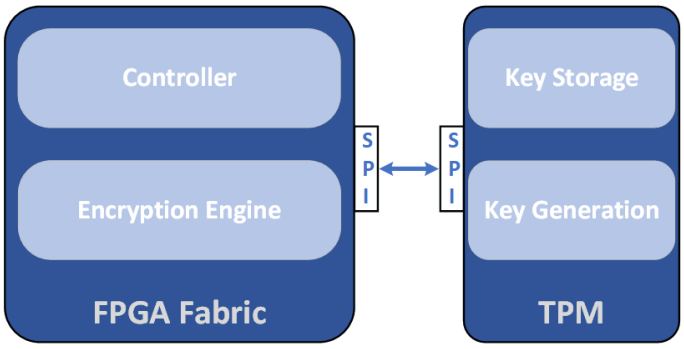

Figure 5.2: Integration of FPGA fabric and TPM.

Figure 5.2 shows the integration of FPGA design fabric with the encryption (AES) engine and the controller over the SPI interface with TPM for the secure key generation and storage. The controller coordinates the co-work between the FPGA fabric and the TPM throughout the encryption process, including communication with the TPM and key update. All the keys in the key list are generated by the built-in TRNG and stored in the tamper-resistant NVM on the TPM chip. All the data communications between the FPGA fabric and the TPM chip, including instruction transmission

and key exchange, are realized through the SPI interface.

Figure 5.3 shows the experimental setup of TPM integration on the Sakura-X board. TPM supports several interfaces such as Low Pin Count (LPC), Serial Peripheral Interface (SPI), and I2C. The TPM chip used in this work is Infineon OPTIGATM TPM 2.0 SLB9670 which is encapsulated in Iridium 9670 Evaluation Boards [76] and connected with the Sakura-X board via SPI port.

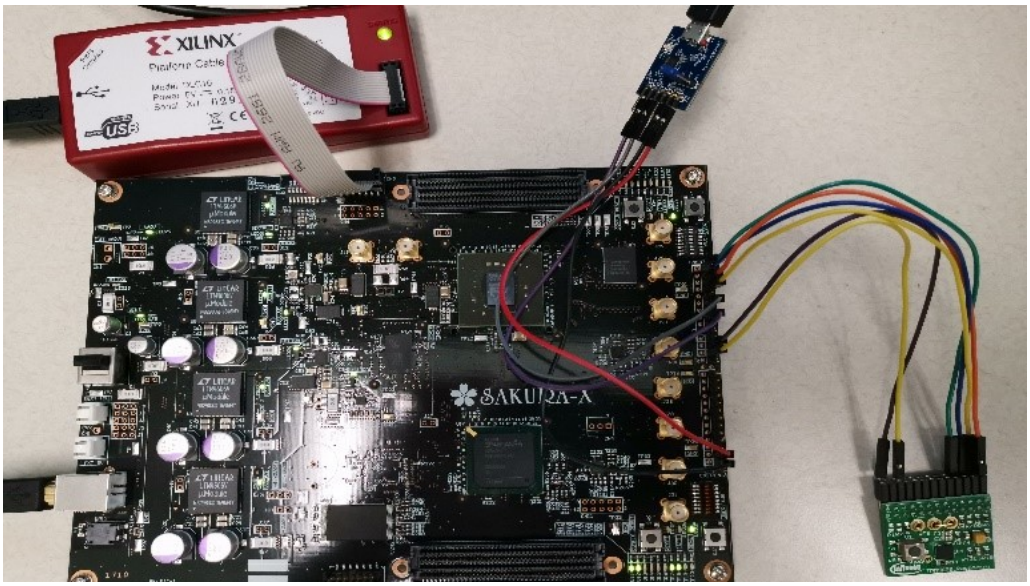

Figure 5.3: TPM configuration on the Sakura-X board.

The high quality of random bit-sequences generated by the TRNG on the TPM chip has been proved by the National Institute of Standards and Technology (NIST) test in the previous work [77]. To utilize the functionality of random key generation with TRNG, the TPM chip is integrated with the MicroBlaze soft processor core which is implemented on the Kintex-7 FPGA chip on the Sakura-X board. A software driver was written to provide the support for integration of TPM with the MicroBlaze software processor [78].

5.3 Experimental Result

5.3.1 Result of CPA Attack Without Key Update Scheme

As discussed in Section 5.1, the proposed key update scheme requires multiple keys. In this experiment, four different keys are picked to generate the key list. For each AES key, we collected 30000 power traces and applied the CPA attack on the first subkey of each key. To reduce the variance caused by variables, we used the same set of random plaintexts in encryption for each key.

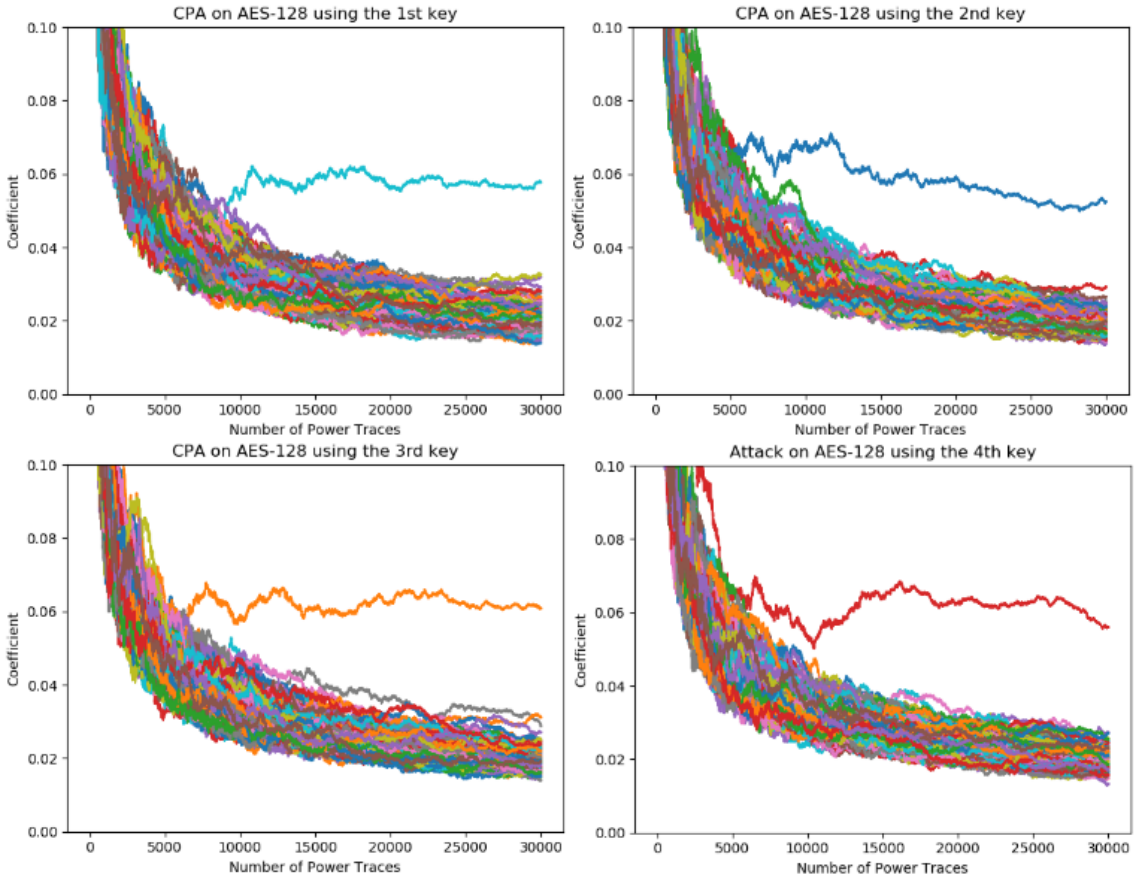

Figure 5.4: The result of CPA attack on the first subkey used in the AES encryption with four different keys.

Figure 5.4 shows the result of CPA attack on the first subkey used in the AES encryption with four different keys. The first subkey of the 1^{st} key **1D 22 BF 01 AC 77 D9 21 EA 34 15 F5 36 89 10 A2** is revealed correctly with around 7000

power traces, and the first subkeys of the 2^{nd} key **F0 1E D2 3C B4 5A 96 78 09 AF 81 EB 27 CD 1F A9**, the 3^{rd} key **97 45 C3 73 1D AD 77 B1 17 B5 76 F4 5B 4C 1E E0** and the 4^{th} key **2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C** can also be revealed with around 5000, 6000, and 5000 power traces, respectively.

5.3.2 Applying the Proposed Key Update Scheme

To mitigate the risk of the side-channel attack, we applied the proposed key update scheme. Based on the result of CPA attack on AES-128 encryption with different keys, the lowest LNTS is around 5000 and the highest LNTS is around 7000 which is much less than the CEMA attack on EM traces collected with the same key and the same set of plaintexts. (Under ideal conditions, the CEMA attack can reach to the same level of efficiency as CPA [61]) If the proposed key update scheme can mitigate the CPA attack, it must also be efficient on CEMA mitigation. To remove the influence of random deviation, we set the value of the Update Period (UP) to 3000 traces (40% less than the lowest LNTS). The sender begins the encryption process with the first random key, then changes the key to the next one after every 3000 full encryption processes following the loop order which is shown in Figure 5.5.

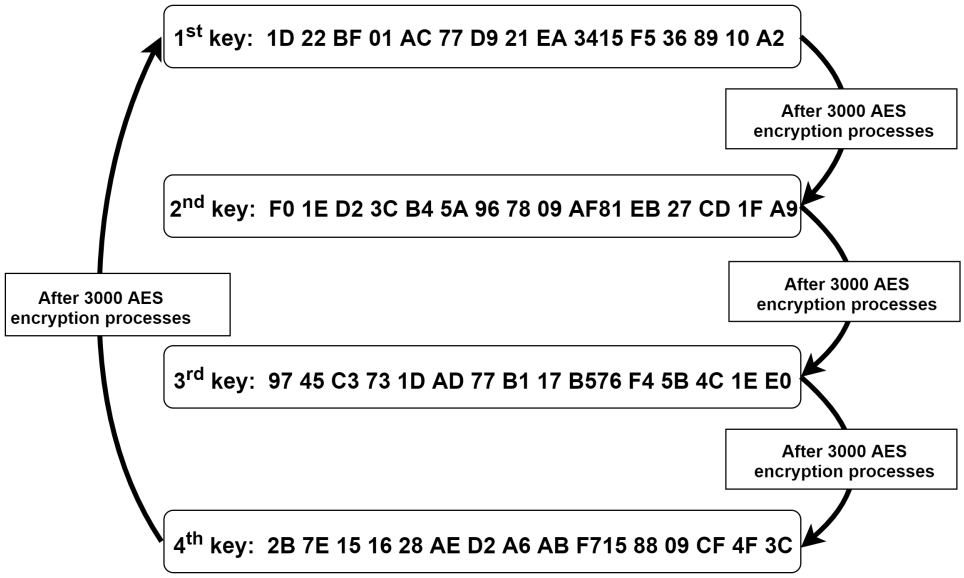

Figure 5.5: Keys and update order.

The receiver also updates the key following the same order shared by the sender for data decryption. We applied the same CPA attack on the collected power traces with the same set of plaintexts using key update scheme to verify the effectiveness. The 1^{st} key was used for encryption in three time periods in this experiment (1^{st}-3000^{th}, 12001^{st}-15000^{th}, 24001^{st}-27000^{th}), it means that totally 9000 power traces using the 1^{st} key are collected for CPA attack. Similarly, the 2^{nd}, 3^{rd} and 4^{th} keys are used for encryption at regular intervals with 9000 (3001^{st}-6000^{th}, 15001^{st}-18000^{th}, 27001^{st}-30000^{th}), 6000 (6001^{st}-9000^{th}, 18001^{st}-21000^{th}) and 6000 (9001^{st}-12000^{th}, 21001^{st}-24000^{th}) random plaintexts, respectively.

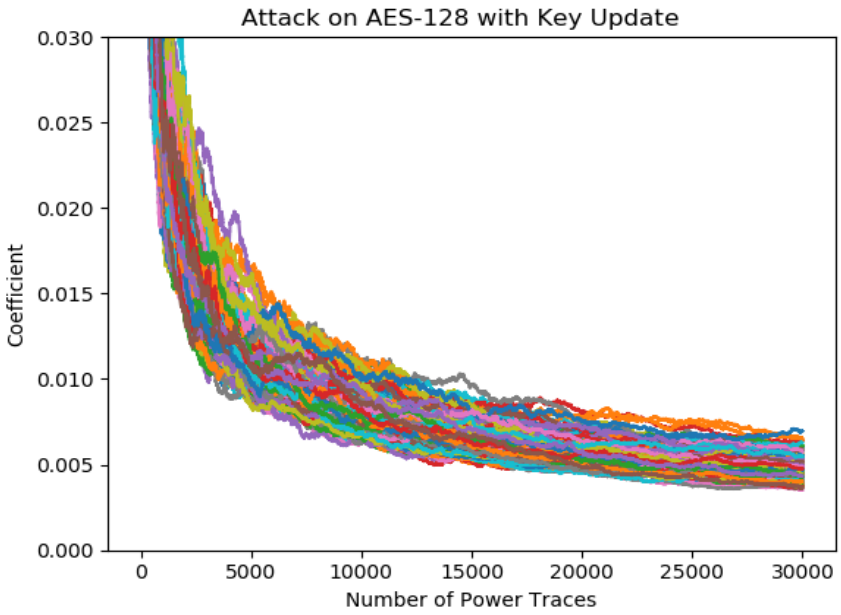

Figure 5.6: The result of CPA attack on the first subkey used in encryption after applying the key update scheme.

The result is shown in Figure 5.6. After applying the proposed key update scheme, none of the subkeys is revealed even with up to 30000 power traces totally, and up to 9000 traces for a single key (the 1^{st} and the 2^{nd} key). In contrast, the first subkeys of the 1^{st} key and the 2^{nd} key can be extracted correctly with around 7000 power traces and 5000 power traces without applying the proposed key update scheme. This means that, even with a deterministic update order, the proposed key update scheme is still secure to mitigate correlation-based attacks because the accumulative correlation model built with previous keys is disturbed continuously every time the new key is applied.

5.3.3 Key Generation on TPM

Figure 5.7 shows the process of key generation on TPM. In this experiment, 8 random keys were generated by the TRNG which can be used for the key update scheme in the encryption process. The average time to generate 8 random keys is 0.014 seconds in 100 runs. The size of the key list and the length of each key are controllable for meeting different security needs.

```
$ ./tpm_keygen.sh
Generating Primary Key
ObjectAttribute: 0x00030072

CreatePrimary Succeed ! Handle: 0x800000ff

Generating Symmertic keys
77D809A16E13C11613F6A2F3F57D3ADD
01662F482BE0BE86C5E142B3541B5FB9
72F0C957178C96ECA600CDB04596F110
2610E9B66AE20A7F4EA7549BA5316F96
9AB4C746E8E898FE992B23BE68B672E5
1EC5BB56BEBF2B6514CF9F88D394BD90
6EC57619896BB1C4F8A9594864049DEB
B7B33CEC5BE58D46680FADEC6413AF40
```

Figure 5.7: The process of key generation on TPM.

5.4 Security Analysis

The encryption engine during execution is vulnerable to side-channel attacks and has been shown in Section 4.4 and 5.3.1. The vulnerability of cryptographic devices roots in the high correlation between the leaked information and the static implementation of the encryption engine. In this work, we propose a key update scheme which is resilient to side-channel attacks.

One advantage of the proposed scheme is that the strength of security is completely controllable by changing the length of the key list, modifying the update order, or adjusting the update period. A longer key list (more random keys) or a higher update frequency (reduce the update period) can enhance the resilience to side-channel attacks further, but also leads to higher overhead. To ensure the security of the proposed scheme, In this work, the sharing process of key list is expected to be performed in a trusted environment before the data communication process.

In [79] [80], Medwed et al. propose a re-keying scheme that generates random keys using a key derivation function. However, the key derivation function is implemented on the same fabric with the encryption engine which brings an extra area overhead and risk of tampering attack. The state-of-the-art FPGA devices natively support key-

rolling to encrypt the bitstream which allows the user to break up the bitstream into multiple AES encryption messages, each encrypted with its own unique rolling key which are derived from the initial key [81]. However, the on-chip AES logic cannot be used for any purpose other than bitstream encryption/decryption and the initial key is stored in the RAM or eFUSE which is still readable by laser stimulation techniques [82]. Moreover, the size of bitstream and the time delay are greatly increased along with activating the key rolling scheme. By comparison, the proposed key update scheme in this work is a general solution scheme and all the keys are generated based on a primary key which is never visible outside of the TPM [83]. In [84], a shifter is used for producing randomness for the key rotation scheme. However, the shifting-based random number generator can only produce pseudo-random numbers. In contrast, we use the built-in TRNG on the TPM to produce true random numbers and the quality has been proved by the NIST test [77]. The use of true random numbers can enhance the strength of key update scheme. To protect the process of key generation, [72] uses a strong PUF to generate keys based on the sub-threshold current array proposed in [85]. The proposed PUF shows good performance but it is only resistant to simple power analysis.

As an efficient countermeasure, masking is widely used for mitigating side-channel attacks. In [40], an order 1 perfectly masked algorithm is presented which masks original secret data with an additively masked value to reduce the correlation between the intermediate values and the input. To increase the efficiency and the security of masking, Threshold Implementation (TI) is proposed in [41] which combines the ideas of secret sharing, threshold cryptography, and multi-party computation protocols. The original secret data is divided into multiple shares using Boolean addition and processed independently, and cannot be revealed unless the number of leaked shares is higher than the preset threshold. However, the area overhead of threshold implementation is very high. For example, the area overhead after applying the

countermeasure based on threshold implementation proposed in [42] is higher than 350%.

In this work, an independent TPM chip is used for key generation. All the keys are generated by the TRNG and stored in the tamper-resistant NVM on the TPM chip so that the risk of tampering attack can be reduced significantly. The area overhead of the proposed design is incurred by the storage for multiple keys used in the key update scheme and depends on the length of each key and the key list. All the keys are stored on the TPM chip, so there is no extra area overhead incurred by key storage on the FPGA fabric. For AES-128, the size of each key is 16 Bytes. The size of NVM on SLB9670 TPM2.0 chip is 6962 Bytes [76] which is able to store up to 435 AES-128 keys. For time overhead, the result shows that the average time to generate one random key is less than 2 milliseconds. Considering the enhancement of security brought by the proposed scheme and the TPM chip, the overhead is fairly small. In addition, TPM supports different sizes and types of keys (RSA, ECC, and AES). This feature makes the proposed scheme more flexible and practical in different scenarios to fulfill various users' needs.

CHAPTER 6: DMA ATTACK AND MITIGATION[4]

6.1 Direct Memory Access (DMA) Attack

6.1.1 Proposed Attack Model

DMA allows peripheral devices to access main system memory independent of the CPU, thereby accelerates the speed of data transfer between external hardwares and the memory. However, it also bring a critical risk to the security. Benefit from the feature of DMA, the attacker can get the direct access to the memory by attaching a new PCIe device. Once the malicious device is connected to the system, the attacker can steal or tamper data stored in the memory without the supervision of CPU.

DMA attack can be performed on different DMA-supported bus standards. Figure 6.1 shows the basic flow of the DMA attack over PCIe.

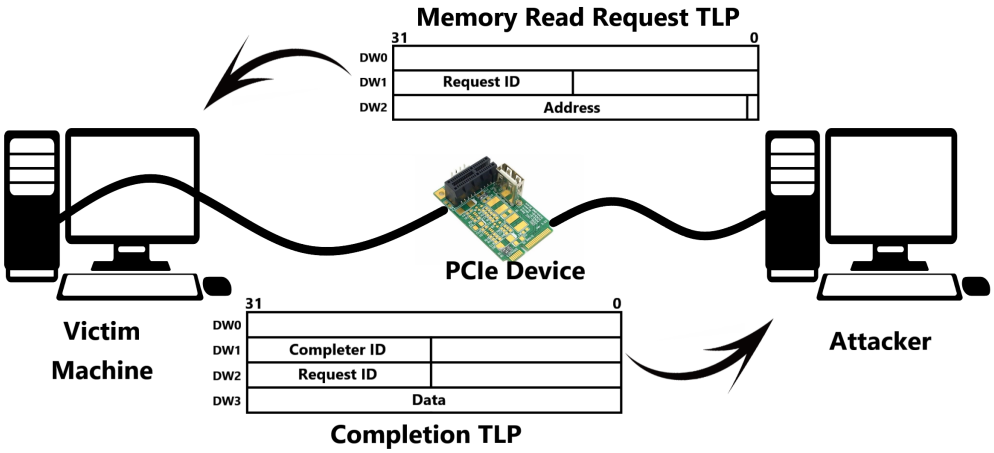

Figure 6.1: Basic DMA attack flow.

As shown in Figure 6.1, once the connection is built, the attacker can send memory read/write requests to the memory on the victim machine using DMA over PCIe. Specifically, there are three steps in a DMA attack:

1. **Access Gain:** To access the main memory, the attacker has to take control of the PCIe device which can be realized by two methods: (1): Compromising the device currently being connected; (2): Inserting a new PCIe device.

2. **Memory Probing:** Before stealing/manipulating data stored in the memory, the attacker traverses the whole live memory to obtain all the available addresses.

3. **Memory Dumping/Manipulation:** After getting all the memory addresses, the attacker can send TLPs to the main memory to dump the data stored in a particular range of addresses, or write malicious data into the memory for further attack.

Moreover, with a loaded kernel module, the attacker is able to perform some more threatening attacks including mounting the file system, spawning system shell on Windows, and removing/changing login password.

In this work, we first perform an attack using a PCIe device to show the vulnerability of DMA attack. The attack model used in this work references the attack model presented in [25] and [86]. The PCIe device is connected to the victim machine via PCIe port and connected to the attacker machine by any of the interfaces (USB, Ethernet, etc.). The attacker controls the PCIe device to send a Memory Read (MRd) request TLP to the victim machine. Once the MRd TLP reaches the PCIe root complex, the victim machine will respond with the completion TLP which contains actual data back to the PCIe device, then back to the attacker. The attacker can either dump all the data stored in the main memory of the victim machine, or a part of data using a specific range of memory addresses.

6.1.2 Experimental Configuration

To configure the attack, we use a PCIe-compatible development board named "NeTV2" with an on-board Xilinx XC7A35T FPGA chip. To improve the appli-

cability of FPGA, Xilinx provides PCIe DMA and PCIe Bridge hard and soft IP blocks for FPGA devices [87], as well as full access to 64-bit memory space without relying on a kernel module running on the victim system. This feature makes FPGA chips compatible with PCIe, but also brings a new risk to the security of the main memory.

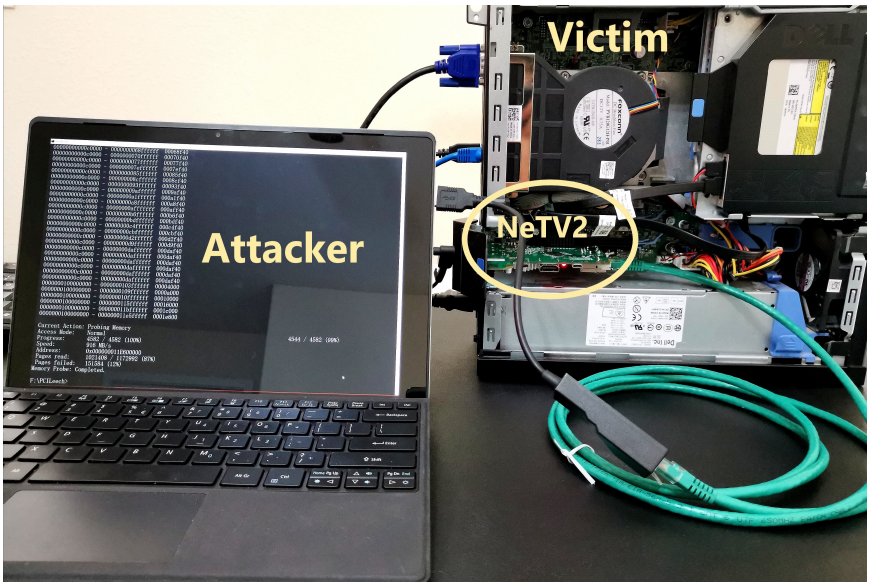

Figure 6.2: Experimental setup of DMA attack.

The setup is shown on Figure 6.2. The NeTV2 board is connected with the victim machine by PCIe port and connected with the attack machine by an Ethernet cable. Since the FPGA natively supports PCIe standard, so there is no need to install any hardware driver on the victim machine. The direct access supported by DMA enables the attacker to control the NeTV2 board to send memory read/write request TLPs to the victim machine, thereby read/write data from/to the main memory on the victim maliciously.

6.1.3 Experimental Result

In this work, the DMA attack is performed on two victim machines with different memory sizes and operating systems. One is running Windows 10 OS with 64GB RAM, and another one is running Ubuntu 18.04 Linux OS with 4GB RAM. In this

section, all the results come from the attack performed on the Windows with 64GB RAM.

```
F:\PCILeech>pcileech probe -device rawudp://ip=192.168.0.222 -v

DEVICE: FPGA: NeTV2 RawUDP PCIe gen2 x1 [0,0,0] [v4.2,0200]
 Memory Map:
 START              END                #PAGES
 0000000000000000 - 000000000009ffff  000000a0
 00000000000c0000 - 000000006bdfffff  0006bd40
 0000000100000000 - 000000010c666fff  0000c667
 000000010c700000 - 00000001156cffff  00008fd0
 0000000115730000 - 000000021fe0efff  0010a6df
 000000021fe40000 - 00000003bdffffff  0019e1c0
 00000003be400000 - 0000000501156fff  00142d57
 00000005011b0000 - 0000000550188fff  0004efd9
 00000005503c7000 - 00000005503d5fff  0000000f
 0000000550400000 - 00000006aa5affff  0015a1b0
 00000006aa800000 - 0000000901ae7fff  002572e8
 0000000901b30000 - 00000009417f6fff  0003fcc7
 0000000941800000 - 0000000aa8ffffff  00167800
 0000000941800000 - 0000000aab975fff  0016a176
 0000000aab998000 - 0000000aea093fff  0003e6fc
 0000000aea0c0000 - 0000000bf9c5ffff  0010fba0
 0000000bfa000000 - 0000000c92ed8fff  00098ed9
 0000000c92ee8000 - 000000108dffffff  003fb118

 Current Action: Probing Memory
 Access Mode:    Normal
 Progress:       67808 / 67808 (100%)
 Speed:          880 MB/s
 Address:        0x000000108E000000
 Pages read:     16748375 / 17358848 (96%)
 Pages failed:   610473 (3%)
Memory Probe: Completed.
```

Figure 6.3: Memory probing on the victim machine.

Figure 6.3 shows the result of memory probing. The NeTV2 board is connected with the attacker's machine, and the static IP of it is 192.168.0.222. In this step, addresses of memory on the victim machine are traversed one by one. Once the process is done, all the readable pages with respective addresses will be acquired by the attacker. As shown in Figure 6.3, the memory on victim machine contains 17358848 pages and 16748375 pages are readable (around 96%) in this case. Besides, the associated address of each page is also shown in this figure clearly.

After getting the full list of readable memory addresses, next step is to perform the attack. By sending the memory read TLP with a particular memory address and receiving the completion TLP, the data stored in that address can be read illegally.

In this work, we dumped all the data stored in the RAM on the victim machine.

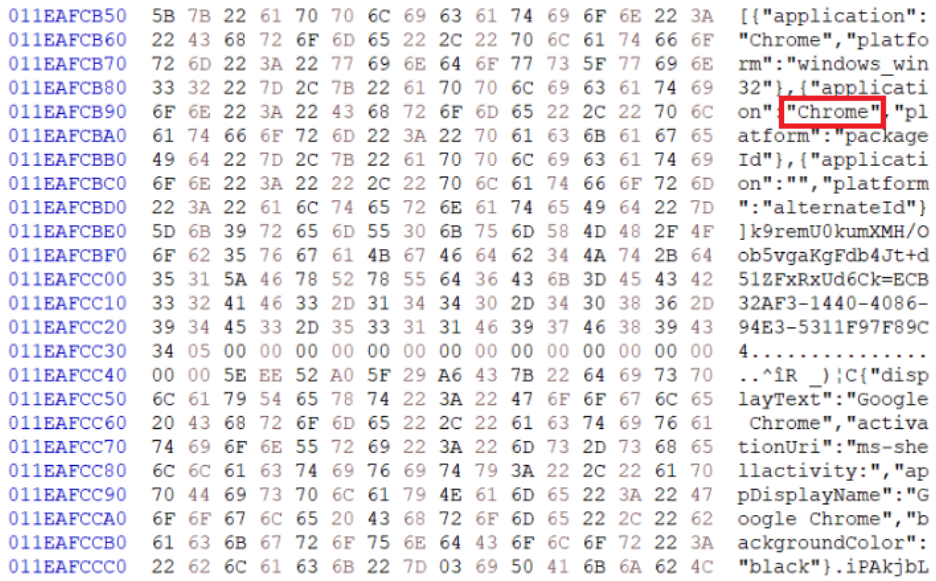

Figure 6.4: A fragment of memory dumped from the victim machine.

Figure 6.4 shows a small part of live memory dumped from the victim machine. The address of each packet is placed on the left side. This memory fragment was being used by the Google Chrome browser when the attacker was dumping the main memory illegally.

Moreover, the unprotected PCIe bus allows the attacker to monitor the live RAM during the run-time and access the file system via a "mounted drive", in this case, the NeTV2 board. In modern computing systems the memory kernel takes charge of memory management, such as storing internal data, buffering data during I/O operations, and sharing memory with other components [88]. Once the kernel module is compromised, the level of security risk will be increased exponentially. Specifically, by inserting various kernel implants into the kernel on the victim machine, it is possible to remove the login password requirement, loading unsigned drivers, executing code and spawn system shells [25] [86].

```
E:\PCILeech>pcileech kmdload -kmd WIN10_X64 -device rawudp://ip=192.168.0.222 -v

DEVICE: FPGA: NeTV2 RawUDP PCIe gen2 x1 [0,0,0] [v4.2,0200]
INFO: PA PT BASE:   0x00000000001ad000
INFO: PA PT:        0x0000000004a04000
INFO: PA HAL HEAP:  0x000000000001098b0
INFO: VA SHELLCODE: 0xfffff7a4c0040210
KMD: Code inserted into the kernel - Waiting to receive execution.
KMD: Execution received - continuing ...
INFO: PA KMD BASE:  0x68ffc000
Kernel reported memory map below:
 START             END               #PAGES
 0000000000001000 - 0000000000057fff  00000057
 0000000000059000 - 000000000009dfff  00000045
 0000000000100000 - 0000000059817fff  00059718
 000000005981a000 - 0000000069807fff  0000ffee
 000000006bdff000 - 000000006bdfffff  00000001
 0000000100000000 - 000000108dffffff  00f8e000
-------------------------------------------------
KMD: Successfully loaded at address: 0x68ffc000
```

Figure 6.5: Loading kernel module.

Fugure 6.5 shows an example of loading kernel module on the victim device via DMA. In this case, the modified kernel module is loaded at address 0x68ffc000. This gives the attacker the ability for accessing the file system of victim machine.

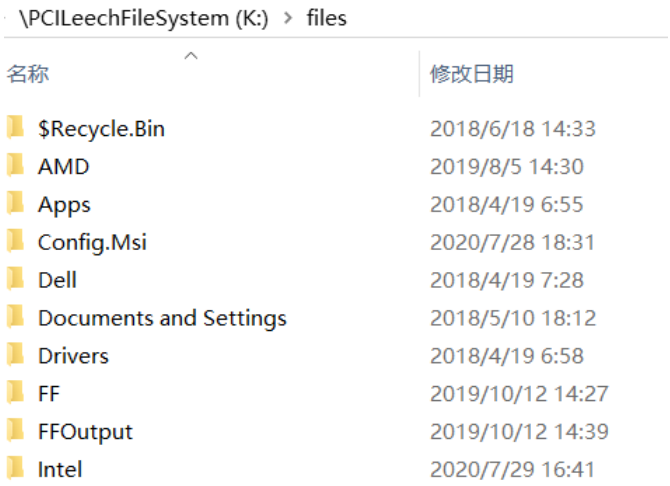

Figure 6.6: Entering the C:\ drive of the victim machine on the attack machine.

Figure 6.6 shows the snapshot from the attacker's machine. After loading the modified kernel module, the file system (C:\ drive in this case) of the victim machine is fully accessible to the attacker and mounted as drive K:\ on the attacker's machine.

The results show the great vulnerability of DMA attack. As a type of non-invasive

side-channel attack, the DMA attack doesn't require any physical modification on the victim machine, so it is hard to defend. What's worse, a recent work shows that the DMA attack can be performed remotely using a WiGig dock [28], which increases the difficulty of detection and defense further.

6.2 DMA Attack Mitigation

Trust On First Use (TOFU) is widely used for building the trusted connection among the host and different nodes. The basic structure of TOFU is simple. Every time a new not-yet-trusted device is connected, the host checks and stores the unique identifier of this device (for example, public identity key or the fingerprint) in its local trusted database. All the devices with identifiers in the database are considered as trusted devices.

The idea of TOFU can be also used for authenticating DMA devices. After applying this mechanism, the device will be blocked unless the host machine can find the unique identifier of it in its trusted database or the authenticated user grants a new permit to this device.

However, as discussed in Section 3.2, there is no unique fingerprint can be used for authenticating PCIe devices currently. The host system uses vendor ID, device ID to recognize different PCIe devices, but cannot distinguish different devices with the same vendor ID and device ID, which makes the access control hard to realize.

To construct a unique identifier for each PCIe device, the profiling time is used in this work. This unique identifier serves as a fingerprinting of each device. Due to the variation in the manufacturing process, each device has a unique physical characteristic such as current flow, IR drop, threshold voltage, and unique delays. These delays affect the response time of the device. Manufacturing variations are also used in creating cryptographic functions such as Physical Unclonable Functions (PUFs).

6.2.1 Proposed Methodology

In this work, a delay-based authentication scheme is proposed for DMA attack mitigation. Figure 6.7 shows the flowchart of the design.

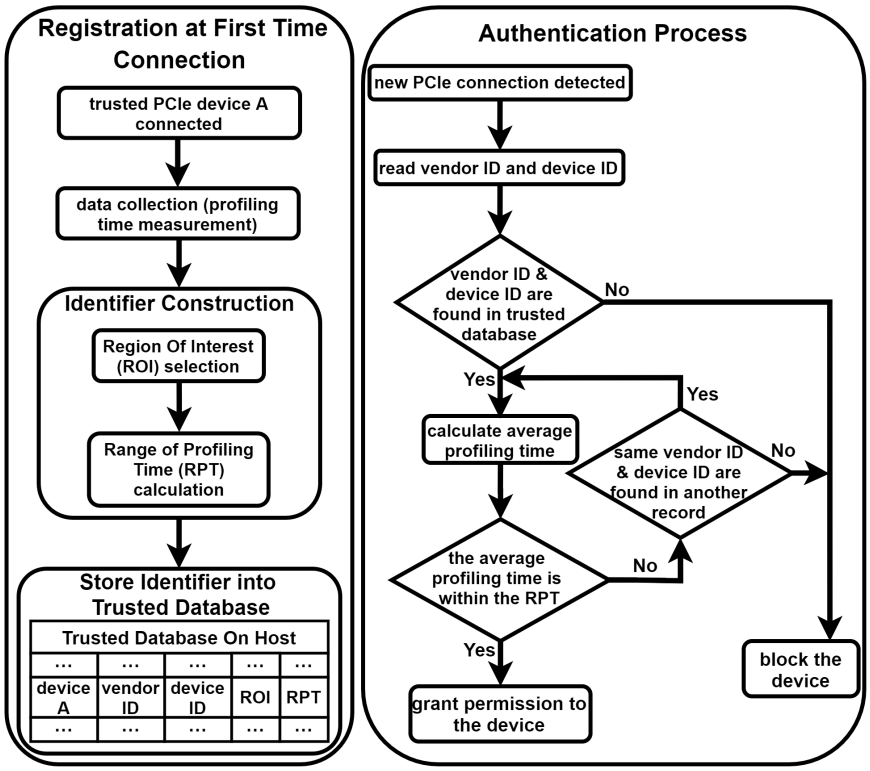

Figure 6.7: Proposed registration and authentication scheme.

As shown in figure, The proposed scheme consists of two processes: registration and authentication. In registration process, the host machine build the unique identifier for each trusted device and stores all the identifiers in its trusted database. These identifiers will be used in the authentication process every time a new PCIe connection is detected.

6.2.1.1 Trusted Device Registration

In the registration process, when a trusted PCIe device is first time connected to the host machine, the authorized user measures the profiling time of this device, constructs an identifier with ROI selection algorithms for this device, and stores the

identifier (Vendor ID, Device ID, Region of Interest (ROI) and Range of Profiling Time (RPT)) into the trusted database.

Before building the trusted database with the profiling time of each device, one issue is the high variance among multiple samples which makes the raw data unusable. This variance is caused by multiple reasons, such as the change of environmental parameters and real-time resource utilization. In this work, we design a data processing framework in registration process to reduce the noise in raw measurements for extracting accurate and stable profiling time.

Data Collection

For each device, multiple sub-datasets are collected, and each sub-dataset contains an equal number of profiling time measurements. Let R denote the number of sub-datasets for each device, and S denotes the number of recorded measurements of profiling time in each sub-dataset. Totally, there are $R \times S$ measurements for each device. After data collection, the raw measurements of profiling time are sorted from lowest to highest in each sub-dataset.

Region Of Interest (ROI) Selection

In data processing, the Region Of Interest (ROI) is a group of samples selected from a dataset for a particular purpose. In this work, the goal of ROI selection is to find the region in the sub-dataset that has the lowest noise. Since the variance in raw measurements is too high, we propose two different algorithms to seek the most valuable ROI in the sorted sub-dataset.

(A) **Difference Based Algorithm**

In the difference based algorithm, we calculate the difference of the same selected regions between each pair of sub-datasets from the first measurement to the last one. One thing to note is that all the sub-datasets are sorted before applying the algorithm. Let L denote the length of the Difference-based ROI (DROI). For

R sub-datasets (each sub-dataset has S measurements of profiling time), there are $C(R, 2)$ different combinations of comparison pairs, and $L(S\text{-}L)$ possible regions.

Algorithm 1: DROI Selection

Input: Number of sub-datasets (R), Number of measurements in each sub-dataset (S), Length of DROI (L)

Output: Difference-based ROI ($DROI$)

$Res[] \leftarrow$ empty list;

for $i \leftarrow 1$ **to** $(S - L)$ **do** `// for all the possible regions`

  for $j \leftarrow 1$ **to** $(R - 1)$ **do**

    for $k \leftarrow (j + 1)$ **to** R **do** `// for all the possible pair combinations`

      calculate AD (absolute difference value between sub-dataset$[j][i : i + L]$ and sub-dataset$[k][i : i + L]$);

      TD (cumulative difference for region i) $\leftarrow TD + AD$;

    end

  end

  $Res[i] \leftarrow TD$;

  $TD \leftarrow 0$;

end

$DROI \leftarrow [d : d + L]$ where d is the index of the minimum value in $Res[]$;

Algorithm 1 shows the process of DROI selection. The inputs are sorted sub-datasets and the length (L) of the DROI. In $Res[]$, the value of element with index i represents the cumulative difference calculated with the particular region $Res[i : i + L]$. In other words, the noise in the region $Res[i : i + L]$ that has the lowest difference with all other sub-datasets is the lowest and considered as the DROI. Note that, the length of DROI is flexible and can be changed according

to different demands.

(B) **Correlation Coefficient Based Algorithm**

Correlation coefficient is a statistical index used to measure the dependency of two variables. In this work, we compute the Pearson's correlation coefficient of all the regions from each pair of sorted sub-datasets to find the Correlation-based ROI (CROI).

Algorithm 2: CROI Selection

Input: Number of sub-datasets (R), Number of measurements in each sub-dataset (S), Length of CROI (L)

Output: Correlation-based ROI ($CROI$)

$Res[] \leftarrow$ empty list;

for $i \leftarrow 1$ **to** $(S - L)$ **do** `// for all the possible regions`

  for $j \leftarrow 1$ **to** $(R - 1)$ **do**

    for $k \leftarrow (j + 1)$ **to** R **do** `// for all the possible pair combinations`

      calculate CC (correlation coefficient between sub-dataset$[j][i : i + L]$ and sub-dataset$[k][i : i + L]$);

      TC (cumulative correlation coefficient for region i) $\leftarrow$ $TC + CC$;

    end

  end

  $Res[i] \leftarrow TC$;

  $TC \leftarrow 0$;

end

$CROI \leftarrow [d : d + L]$ where d is the index of the maximum value in $Res[]$;

Algorithm 2 shows the process of CROI selection. The inputs are totally the

same as DROI selection. However, in DROI selection, the lowest value in result list Res[] is used because the lowest difference means the same selected regions from each pair of sub-datasets have the highest similarity (in other words, the lowest noise). However, in CROI selection, the highest value in result list Res[] is selected because the highest coefficient represents the highest similarity.

At the end of the selection process, the overlapped region between the DORI and CROI is selected as the ROI for identifier construction.

Construct and store identifier

The process of ROI selection figures out the region with the lowest level of noise. By calculating the average profiling time of all the measurements in ROI for each sub-dataset and combining the results of all the sub-datasets, we get the Range of Profiling Time (RPT) which can be used as a part of the identifier. The ROI and RPT will be stored in the trusted database on the host machine, along with the vendor ID and device ID of this trusted device.

6.2.1.2 Authentication

As shown in Figure 6.7, in authentication process, every time a new PCIe connection is detected, the system reads the vendor ID and device ID of this PCIe device. Once the vendor ID and device ID are found in the trusted database, the host machine will collect a number of profiling time measurements and calculate the average profiling time based on the ROI stored in the same record. If the average profiling time of this device is within the RPT stored in the same record, the permission will be granted to the device, otherwise the system will recheck for other records that contain the same vendor ID and device ID (in case of more than one device of the same model are registered) until all the records have been traversed.

After all the records in the trusted database are traversed but the new connected device doesn't match any record of registration, the system will send a warning and

block it until this device is registered by the authenticated user or unplugged.

6.2.2 Experimental Setup and Result

In this work, three TL-WN881ND wireless PCIe adapters (called device A, device B, and device C) from TP-LINK [89] that have the same properties (same vendor ID and device ID) are used for verifying the proposed design. In data collection, the elapsed time of reading configuration space on the PCIe device is measured as profiling time.

In experiments, we measured the profiling time of each device 10000 times and repeated this process 30 times under the same conditions. After data collection, we have 30 sub-datasets and each sub-dataset has 10000 measurements of profiling time for each device.

Figure 6.8 shows all 10000 measurements of profiling time in one sub-dataset collected from device A, and Figure 6.9 shows all 30 sub-datasets with sorted measurements collected from device A (different sub-datasets have different colors). For comparison across different devices, Figure 6.10 plots all the sorted sub-datasets collected from all three devices (sub-datasets collected from the same device has the same color).

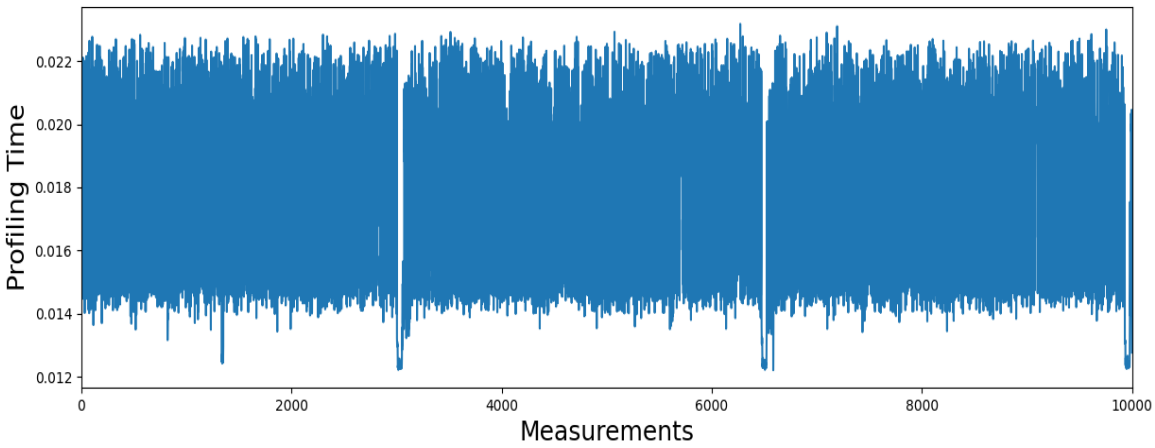

Figure 6.8: All 10000 measurements of profiling time in one sub-dataset collected from device A.

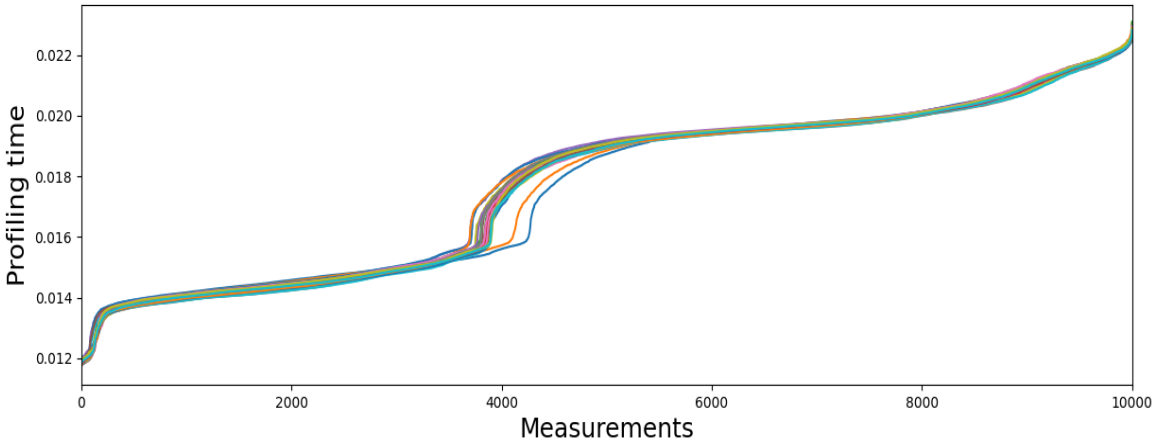

Figure 6.9: Comparison of all 30 sorted sub-datasets collected from device A.

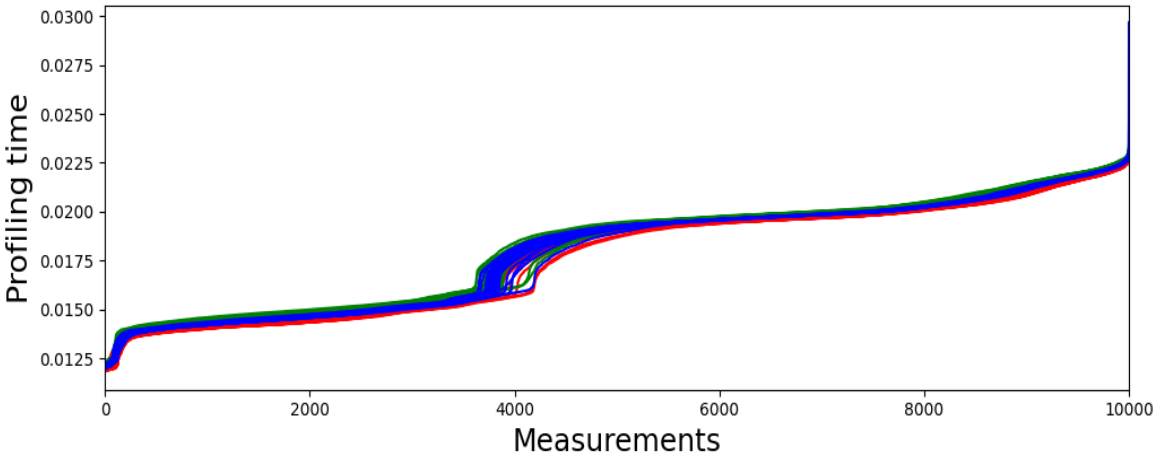

Figure 6.10: All the sub-datasets collected from all three devices. (red, green and blue colors represent sub-datasets collected from device A, B and C, respectively.

As shown in Figure 6.8 and Figure 6.9, the variance among the collected measurements is very high (from 12 ms to 23 ms), and the difference among different sub-datasets is also very high in some regions (between the 3500^{th} point and the 5500^{th} point). In Figure 6.10, it is clear to see that the difference of profiling time measurements collected from different devices is small. Moreover, Some samples collected from device device C are even higher than 25ms due to the environmental noise. Accordingly, it is difficult to construct identifiers without data processing.

Then, we applied the two proposed algorithms of ROI selection on the raw mea-

surements collected from device A. In this case, we set the length of DROI and CROI to 3000. Figure 6.11 and Figure 6.12 show the result of DROI selection and CROI selection.

```
Min Difference:  0.02302205618222555
Difference based ROI (DROI):  5464 - 8464
>>>
Max correlation:  434.7485805839838
Correlation based ROI (CROI):  6991 - 9991
>>>
```

Figure 6.11: The output of DROI selection and CROI selection.

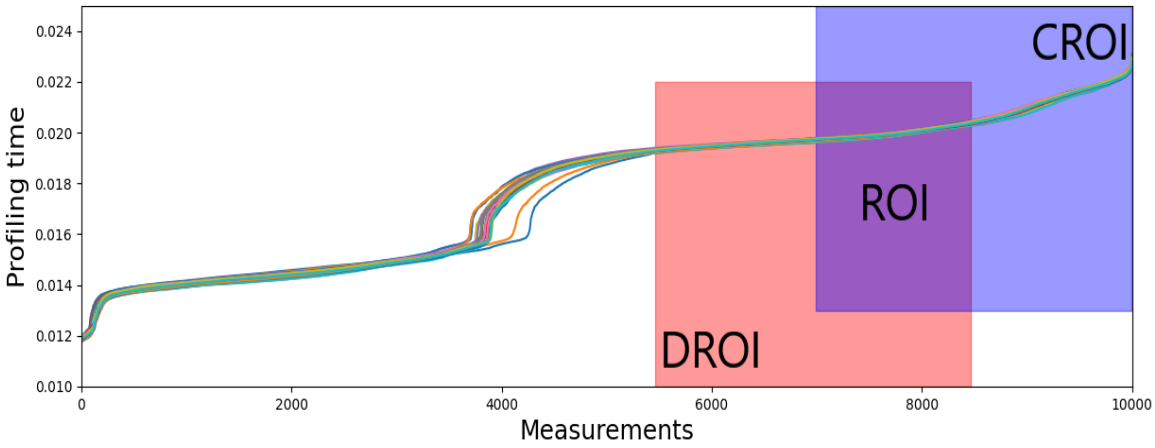

Figure 6.12: Result of DROI selection and CROI selection.

In this experiment, the lengths of both CROI and DROI are set to 3000. The result shows that the DROI of device A is from the 5464^{th} (included) point to the 8464^{th} point (not included) in each sorted sub-dataset, and the CROI is from the 6991^{st} point (included) to the 9991^{st} point (not included). We selected the overlapped region of DROI and CROI as ROI (6991^{st} (included) - 8464^{th} (not included)). This region has the lowest noise and the highest stability because samples in this region from data collections over different periods (sub-datasets in this case) have both the lowest difference and the highest correlation with each other. Moreover, based on the result of comparison among three devices, we noticed that all the devices of the same type

have very similar ROIs with each other, so we used the ROI of device A for all three devices to construct the identifier.

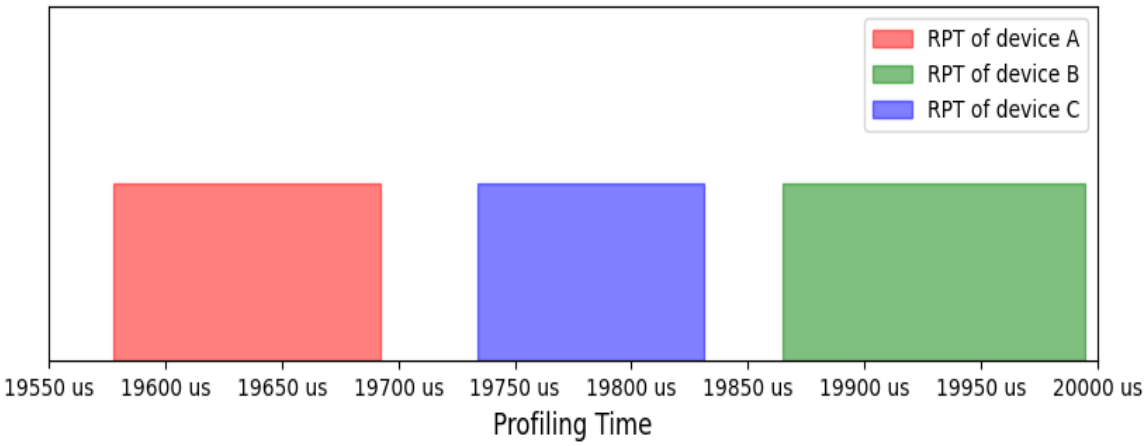

Figure 6.13: Range of Profiling Time (RPT) of each device.

For each device, we calculated the average profiling time of all the measurements in ROI in each sub-dataset. The average profiling times of all the sub-datasets were combined to form the RPT for each device. As shown in Figure 6.13, RPTs range from 19577.90 us to 19692.26 us, 19865.05 us to 19994.50 us, 19734.17 us to 19831.31 us for device A, B, and C, respectively. There is no overlapped area among the three devices, which means that the RPT of each device is unique. RPTs were used as identifiers of each device and stored in the trusted database with vendor ID, device ID, and ROI on the host machine for future authentication.

In authentication, each time we collected 10000 measurements of profiling time and calculated the average profiling time of all the measurements in the ROI. This process was repeated 50 times for each device under the same conditions. The result is shown in Table 6.1.

Table 6.1: Success Rate of Authentication

Devices	Number of Hit (Calculated Average Profiling Time is Within RPT)	Number of Miss (Calculated Average Profiling Time is Outside RPT)	Hit Rate
Device A	46	4	92%
Device B	49	1	98%
Device C	45	5	90%

As shown in Table 6.1, the proposed design has a high success rate in authentication. Device B is authenticated with 98% success rate in 50 authentication processes and the average success rate of all the devices is higher than 93%. Based on the experimental result, the proposed scheme is proved to be practical.

6.2.3 Security Analysis

The proposed framework uses profiling time to construct identifiers for PCIe devices. In comparison with other existing mitigation countermeasures, the delay-based authentication model has two major advantages:

- The proposed design does not require any hardware-level or protocol-level modification. Existing countermeasures to DMA attacks, such as [13] and [90], need either an adjustment to the current protocol or physical modification of devices that make designs less practical in the real world. In this work, the authentication is achieved by extracting the time-delay characteristic in device profiling and no additional change on the hardware is needed, therefore it is more feasible as compared to other existing works.

- The cost of the proposed design is low. IOMMU has been proved as an efficient countermeasure to DMA attacks, but activating IOMMU will reduce the real-time performance of computing systems significantly [14]. As a contrast, the

delay-based authentication model presented in this work is lightweight. The registration is a one-time process and there is no impact on performance after the authentication is done.

There are two types of overhead in this design. The first one is the resource overhead which is caused by the storage of the trusted database. In the real world, the PCIe devices connected or has been connected to one host machine are fewer therefore the size of the trusted database is fairly small. For example, if the trusted database has 30 records, the overall overhead of storage is only 1.81 KB.

Table 6.2: Time Overhead (One Device)

Registration Overhead (in minutes)		Authentication Overhead (in minutes)	
Measuring	Identifier Construction	Measuring	Calculation
100 mins	15.7 mins	3.4 mins	4e-7 mins

Another overhead is the time overhead which is shown in Table 6.2. In this design, there are two kinds of time overhead: registration overhead and authentication overhead. For each device, we collected 300000 measurements in the registration process which took 100 minutes and the process of identifier construction took 15.7 minutes. For authentication, 10000 measurements were collected which took 3.4 minutes and the calculation of average profiling time took 4e-7 minutes. The registration process takes longer, but it is just a one-time process. In comparison, the time overhead of authentication is 3.4. It is important to note that, the time overhead of identifier construction depends on the performance of the host machine. In this experiment, for both the registration process and authentication process, the CPU used is Core i5-3470.

In this work, we use a Python-based interface for PCIe and the profiling time is measured using the time library of Python. As shown in Figure 6.8, the variance among different raw measurements of profiling time is very high. Even the proposed

design is able to seek out the ROI with the lowest noise and construct unique identifiers for different devices, the lack of high accuracy brings three major issues:

- **Overlapping**. As shown in Figure 6.13, there is no overlapped area. However, if the number of devices increases, there might be some areas of overlap which makes each identifier not unique anymore. Assuming the devices from the same vendor and the same family will remain fewer per PCIe interface, this proposed scheme works.

- **High overhead of registration time**. The higher the accuracy of measurement, the less quantity of samples for identifier construction is needed. If the accuracy of measurement can be increased, the registration process will need fewer measurements for constructing identifiers therefore the time overhead will be also reduced.

- **Insufficient success rate of authentication**. As shown in Table 6.1, the average success rate of authentication is higher than 93%. However, to apply the proposed scheme in the real world, the success rate must be increased to a higher level which can be realized by improving the accuracy of measurement.

In order to avoid overlapping of RPTs and reduce the time overhead of registration, further research will focus on improving the accuracy of measurement which can be realized by two methods:

- **Noise elimination**. By applying mathematical models in measurement such as straight line fitting [91], the noise caused by systematic errors can be eliminated or reduced to the lowest level.

- **Logic Analyzer**. Benefit from the high resolution, the logic analyzer can capture and display signals much more precisely than the clock integrated in the host machine.

In addition, applying classification algorithms can be also valuable to reduce the number of measurements required in the authentication process, such as Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN).

CHAPTER 7: CONCLUSIONS

With the rapid development of digitalization in manufacturing and communication, it also comes with critical challenges on security and privacy, such side-channel attacks. In order to fulfill the demand for security enhancement, this work presents a full-detailed exploration on multiple types side-channel attacks, and proposes feasible and lightweight countermeasures for mitigating the risk of correlation-based analysis and the DMA attack.

In the aspect of power/EM based attacks, this dissertation performs template attack and correlation-based analysis on both software-based and hardware-based implementations, and gives a comprehensive comparison of efficiency and strength among different attack models. Moreover, this work also proposes a key update scheme as a countermeasure for correlation-based analysis. The keys are updated during short intervals to eliminate the side channel analysis by calculating the Least Needed power/EM Traces (LNT) of the target device and updating the key before any subkey can be revealed on each node synchronously, therefore the risk of the power analysis attack and the EM analysis attack is mitigated significantly as shown in experiments. In the proposed framework, all the keys are generated and stored securely on the TPM, and the security is controllable to meet different demands by changing the length of each key and the number of keys.

In the aspect of DMA attack, this dissertation first performs a successful attack on the main memory of the victim machine, then proposes a lightweight authentication scheme for DMA-supported PCIe devices based on the unique identifiers constructed with the profiling time. By applying the proposed ROI selection algorithms, the noise in measurements is reduced remarkably. The result of experiments shows that

there is no overlapped area among RPTs of three PCIe devices and the success rate of authentication is greater than 93%. The proposed design does not require any modifications to hardware and protocol, and does not have any negative effect on the performance of computing systems, make this design more feasible than any other existing countermeasures.

REFERENCES

[1] Y. Gui, A. S. Siddiqui, S. M. Tamore, and F. Saqib, "Investigation of vulnerabilities on smart grid end devices," in *2019 IEEE CyberPELS (CyberPELS)*, pp. 1–6, 2019.

[2] Y. Gui, A. S. Siddiqui, S. M. Tamore, and F. Saqib, "Security vulnerabilities of smart meters in smart grid," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 3018–3023, 2019.

[3] Y. Gui, S. M. Tamore, A. S. Siddiqui, and F. Saqib, "Key update countermeasure for correlation-based side-channel attacks," *Journal of Hardware and Systems Security*, vol. 4, p. 167–179, 2020.

[4] Y. Gui, A. S. Siddiqui, G. S. Nicholas, M. Hughes, and F. Saqib, "A lightweight delay-based authentication scheme for dma attack mitigation," in *2021 22st International Symposium on Quality Electronic Design (ISQED)*, 2021.

[5] A. Stefanov and C. Liu, "Cyber-power system security in a smart grid environment," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–3, 2012.

[6] M. Smith and E. Lostri, "The hidden costs of cybercrime," 2020. Available at `https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf`.

[7] P. Nohe, "How strong is 256-bit encryption?," 2019. Available at `https://www.thesslstore.com/blog/what-is-256-bit-encryption/`.

[8] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99* (M. Wiener, ed.), (Berlin, Heidelberg), pp. 388–397, Springer Berlin Heidelberg, 1999.

[9] D. J. Bernstein, "Cache-timing attacks on aes," Citeseer, 2005.

[10] Microsoft, "Bitlocker countermeasures," 2019. Available at `https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-countermeasures`.

[11] D. Abramson and G. Neiger, "Intel virtualization technology for directed i/o," *Intel Technology Journal*, vol. 10, 2006.

[12] AMD, "Amd i/o virtualization technology (iommu) specification," 2020. Available at `https://www.amd.com/en/support/tech-docs/amd-io-virtualization-technology-iommu-specification`.

[13] N. Edwards, T. Koulouris, and M. KrauseM, "Pcie component authentication," 2019. Available at `https://pcisig.com/pcie%C2%AE-component-authentication"`.

[14] M. Ben-Yehuda, J. Xenidis, M. Ostrowski, K. Rister, A. Bruemmer, and L. Van Doorn, "The price of safety: Evaluating iommu performance," in *The Ottawa Linux Symposium*, pp. 9–20, 2007.

[15] S. Schonhart, A. Muller, L. Boszormenyi, and S. Podlipnig, "Des challenge." Available at `http://cs-exhibitions.uni-klu.ac.at/index.php?id=263`.

[16] S. Neves, "How long does it take to crack des and aes?," 2011. Available at `https://crypto.stackexchange.com/questions/752/how-long-does-it-take-to-crack-des-and-aes`.

[17] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology — CRYPTO '96* (N. Koblitz, ed.), (Berlin, Heidelberg), pp. 104–113, Springer Berlin Heidelberg, 1996.

[18] NewAE, "Template attacks," 2018. Available at `https://wiki.newae.com/Template_Attacks`.

[19] T. Kubota, K. Yoshida, M. Shiozaki, and T. Fujino, "Deep learning side-channel attack against hardware implementations of aes," in *2019 22nd Euromicro Conference on Digital System Design (DSD)*, pp. 261–268, 2019.

[20] L. Zhang, X. Xing, J. Fan, Z. Wang, and S. Wang, "Multi-label deep learning based side channel attack," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2020.

[21] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004* (M. Joye and J.-J. Quisquater, eds.), (Berlin, Heidelberg), pp. 16–29, Springer Berlin Heidelberg, 2004.

[22] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side-channel(s)," in *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, (Berlin, Heidelberg), p. 29–45, Springer-Verlag, 2002.

[23] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems — CHES 2001* (Ç. K. Koç, D. Naccache, and C. Paar, eds.), (Berlin, Heidelberg), pp. 251–261, Springer Berlin Heidelberg, 2001.

[24] A. Bu, W. Dai, M. Lu, H. Cai, and W. Shan, "Correlation-based electromagnetic analysis attack using haar wavelet reconstruction with low-pass filtering on an fpga implementaion of aes," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 1897–1900, 2018.

[25] U. Frisk, "Direct memory attack the kernel," 2016. Available at https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEF%20CON%2024%20-%20Ulf-Frisk-Direct-Memory-Attack-the-Kernel.pdf.

[26] PCI-SIG, "Specifications," 2021. Available at https://pcisig.com/specifications.

[27] XILLYBUS, "Down to the tlp: How pci express devices talk," 2012. Available at http://xillybus.com/tutorials/pci-express-tlp-pcie-primer-tutorial-guide-1.

[28] Eclypsium, "Direct memory access - a walk down memory lane," 2020. Available at https://eclypsium.com/2020/01/30/direct-memory-access-attacks.

[29] B. Ruytenberg, "When lightning strikes thrice: Breaking thunderbolt 3 security," 2020. Available at https://thunderspy.io.

[30] P. Sasdrich, A. Moradi, and T. Güneysu, "Hiding higher-order side-channel leakage," in *Topics in Cryptology – CT-RSA 2017* (H. Handschuh, ed.), (Cham), pp. 131–146, Springer International Publishing, 2017.

[31] N. Mentens, B. Gierlichs, and I. Verbauwhede, "Power and fault analysis resistance in hardware through dynamic reconfiguration," in *Cryptographic Hardware and Embedded Systems – CHES 2008* (E. Oswald and P. Rohatgi, eds.), (Berlin, Heidelberg), pp. 346–362, Springer Berlin Heidelberg, 2008.

[32] J.-S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Cryptographic Hardware and Embedded Systems* (Ç. K. Koç and C. Paar, eds.), (Berlin, Heidelberg), pp. 292–302, Springer Berlin Heidelberg, 1999.

[33] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, 2010.

[34] C. Wang, M. Yan, Y. Cai, Q. Zhou, and J. Yang, "Power profile equalizer: A lightweight countermeasure against side-channel attack," in *2017 IEEE International Conference on Computer Design (ICCD)*, pp. 305–312, 2017.

[35] J. Danger, S. Guilley, S. Bhasin, and M. Nassar, "Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors," in *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, pp. 1–8, 2009.

[36] 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*, pp. 403–406, 2002.

[37] D. D. Hwang, K. Tiri, A. Hodjat, B. . Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "Aes-based security coprocessor ic in 0.18-*muhboxm* cmos with resistance to differential power analysis side-channel attacks," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, 2006.

[38] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography in the standard model." Cryptology ePrint Archive, Report 2008/240, 2008. Available at `https://eprint.iacr.org/2008/240`.

[39] S. Dziembowski and K. Pietrzak, "Intrusion-resilient secret sharing," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 227–237, 2007.

[40] J. Blömer, J. Guajardo, and V. Krummel, "Provably secure masking of aes," in *Selected Areas in Cryptography* (H. Handschuh and M. A. Hasan, eds.), (Berlin, Heidelberg), pp. 69–83, Springer Berlin Heidelberg, 2005.

[41] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Information and Communications Security* (P. Ning, S. Qing, and N. Li, eds.), (Berlin, Heidelberg), pp. 529–545, Springer Berlin Heidelberg, 2006.

[42] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of aes," in *Advances in Cryptology – EUROCRYPT 2011* (K. G. Paterson, ed.), (Berlin, Heidelberg), pp. 69–88, Springer Berlin Heidelberg, 2011.

[43] G. Agosta, A. Barenghi, and G. Pelosi, "A code morphing methodology to automate power analysis countermeasures," in *DAC Design Automation Conference 2012*, pp. 77–82, 2012.

[44] Y. N. Srikant and P. Shankar, *Compiler Design Handbook: Optimizations and Machine Code Generation*. USA: CRC Press, Inc., 2002.

[45] T. C. Group, "Trusted platform module (tpm) summary." Available at `https://trustedcomputinggroup.org/resource/trusted-platform-module-tpm-summary`.

[46] T. Markettos, C. Rothwell, B. Gutstein, A. Pearce, P. Neumann, S. Moore, and R. Watson, "Thunderclap: Exploring vulnerabilities in operating system iommu protection via dma from untrustworthy peripherals," in *2019 Network and Distributed System Security Symposium*, 2019.

[47] D. Wendlandt, D. Andersen, and A. Perrig, "Perspectives: Improving ssh-style host authentication with multi-path probing," in *USENIX Annual Technical Conference*, 2008.

[48] M. O. Choudary and M. G. Kuhn, "Efficient, portable template attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 490–501, 2018.

[49] A. Chakraborty, Y. Xie, and A. Srivastava, "Template attack based deobfuscation of integrated circuits," in *2017 IEEE International Conference on Computer Design (ICCD)*, pp. 41–44, 2017.

[50] G. Fan, Y. Zhou, H. Zhang, and D. Feng, "How to choose interesting points for template attacks more effectively?," in *Trusted Systems* (M. Yung, L. Zhu, and Y. Yang, eds.), (Cham), pp. 168–183, Springer International Publishing, 2015.

[51] J. Brownlee, "An introduction to feature selection," 2020. Available at `https://machinelearningmastery.com/an-introduction-to-feature-selection`.

[52] Gearbest, "Magic blue uu bluetooth bulb - rgbw e27." Available at `https://www.gearbest.com/smart-light-bulb/pp_230349.html`.

[53] Bluetooth, "New bluetooth standard built for the universal remote control," 2008. Available at `https://www.bluetooth.com/news/pressreleases/2008/07/22/new-bluetoothstandard-built-for-the-universal-remote-control`.

[54] Bluetooth, "Bluetooth radio versions." Available at `https://www.bluetooth.com/learn-about-bluetooth/radio-versions/`.

[55] U. Shaked, "Reverse engineering a bluetooth lightbulb," 2016. Available at `https://urish.medium.com/reverse-engineering-a-bluetooth-lightbulb-56580fcb7546`.

[56] Realtek, "Realtek rtl8762ar/ag/aj/ak-cg datasheet 1.3," 2017. Available at `https://vrtp.ru/index.php?act=Attach&type=post&id=742189`.

[57] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002* (B. S. Kaliski, ç. K. Koç, and C. Paar, eds.), (Berlin, Heidelberg), pp. 13–28, Springer Berlin Heidelberg, 2003.

[58] ChipWhisperer, "chipwhisperer," 2021. Available at `https://github.com/newaetech/chipwhisperer`.

[59] T. Messerges, E. Dabbish, and R. Sloan, "Investigations of power analysis attacks on smartcards," 09 1999.

[60] O. Lo, W. J. Buchanan, and D. Carson, "Power analysis attacks on the aes-128 s-box using differential power analysis (dpa) and correlation power analysis (cpa)," *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 88–107, 2017.

[61] Y. Nomata, M. Matsubayashi, K. Sawada, and A. Satoh, "Comparison of side-channel attack on cryptographic cirucits between old and new technology fpgas," in *2016 IEEE 5th Global Conference on Consumer Electronics*, pp. 1–4, 2016.

[62] S. Morioka and A. Satoh, "An optimized s-box circuit architecture for low power aes design," in *Cryptographic Hardware and Embedded Systems - CHES 2002* (B. S. Kaliski, ç. K. Koç, and C. Paar, eds.), (Berlin, Heidelberg), pp. 172–186, Springer Berlin Heidelberg, 2003.

[63] S. Laboratory, "Sakura-x," 2014. Available at `http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-X.html`.

[64] R. B. INC, "Lna-1050," 2006. Available at `http://www.rfbayinc.com/upload/files/lna/lna-1050.pdf`.

[65] Tektronix, "Dpo/dsa/mso70000 series oscilloscopes," 2013. Available at `http://download.tek.com/document/55W-22447-9.pdf`.

[66] ChipWhisperer, "Cw505 planar h-field probe," 2018. Available at `https://wiki.newae.com/CW505\_Planar\_H-Field\_Probe`.

[67] S. Laboratory, "Sasebo-gii," 2012. Available at `http://satoh.cs.uec.ac.jp/SASEBO/en/board/sasebo-g2.html`.

[68] D. Contest, "Dpa contest v2," 2012. Available at `http://www.dpacontest.org/v2/index.php`.

[69] T. C. Group, "Tpm structures," 2011. Available at `https://trustedcomputinggroup.org/wp-content/uploads/TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf`.

[70] Y. Gui, A. S. Siddiqui, and F. Saqib, "Hardware based root of trust for electronic control units," in *SoutheastCon 2018*, pp. 1–7, 2018.

[71] A. S. Siddiqui, Y. Gui, D. Lawrence, S. Laval, J. Plusquellic, M. Manjrekar, B. Chowdhury, and F. Saqib, "Hardware assisted security architecture for smart grid," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2890–2895, 2018.

[72] X. Xi, A. Aysu, and M. Orshansky, "Fresh re-keying with strong pufs: A new approach to side-channel security," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 118–125, 2018.

[73] Y. Jin, "Introduction to hardware security," *Electronics*, vol. 4, no. 4, pp. 763–784, 2015.

[74] K. Rosenfeld and R. Karri, "Attacks and defenses for jtag," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 36–47, 2010.

[75] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J. Seifert, "Key extraction using thermal laser stimulation: A case study on xilinx ultrascale fpgas," 2018.

[76] Infineon, "Iridium slb 9670 tpm2.0 linux." Available at `https://www.infineon.com/cms/en/product/evaluation-boards/iridium9670-tpm2.0-linux`.

[77] A. Suciu and T. Carean, "Benchmarking the true random number generator of tpm chips," 2010.

[78] A. S. Siddiqui, Y. Gui, and F. Saqib, "Secure boot for reconfigurable architectures," *Cryptography*, vol. 4, no. 4, 2020.

[79] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni, "Fresh re-keying: Security against side-channel and fault attacks for low-cost devices," in *Progress in Cryptology – AFRICACRYPT 2010* (D. J. Bernstein and T. Lange, eds.), (Berlin, Heidelberg), pp. 279–296, Springer Berlin Heidelberg, 2010.

[80] M. Medwed, C. Petit, F. Regazzoni, M. Renauld, and F.-X. Standaert, "Fresh re-keying ii: Securing multiple parties against side-channel and fault attacks," in *Smart Card Research and Advanced Applications* (E. Prouff, ed.), (Berlin, Heidelberg), pp. 115–132, Springer Berlin Heidelberg, 2011.

[81] Xilinx, "Using encryption and authentication to secure an ultrascale/ultrascale+ fpga bitstream," 2019. Available at `https://www.xilinx.com/support/documentation/application_notes/xapp1267-encryp-efuse-program.pdf`.

[82] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert, "Key extraction using thermal laser stimulation: A case study on xilinx ultrascale fpgas." Cryptology ePrint Archive, Report 2018/717, 2018. `https://eprint.iacr.org/2018/717`.

[83] T. C. Group, "Endorsement key (ek) and platform certificate enrollment specification frequently asked questions," 2013. Available at `https://trustedcomputinggroup.org/wp-content/uploads/IWG-EK-CMC-enrollment-for-TPM-v1-2-FAQ-rev-April-3-2013.pdf`.

[84] A. Wang, C. Wang, X. Zheng, W. Tian, R. Xu, and G. Zhang, "Random key rotation: Side-channel countermeasure of ntru cryptosystem for resource-limited devices," *Computers & Electrical Engineering*, vol. 63, pp. 220–231, 2017.

[85] M. Kalyanaraman and M. Orshansky, "Novel strong puf based on nonlinearity of mosfet subthreshold operation," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 13–18, 2013.

[86] U. Frisk, "Pcileech," 2020. Available at `https://github.com/ufrisk/pcileech`.

[87] Xilinx, "Pci express and xilinx technology," 2020. Available at `https://www.xilinx.com/products/technology/pci-express.html`.

[88] Microsoft, "Windows kernel-mode memory manager," 2018. Available at `https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/windows-kernel-mode-memory-manager`.

[89] TP-LINK, "Tl-wn881nd," 2020. Available at `https://www.tp-link.com/il/home-networking/adapter/tl-wn881nd/`.

[90] Intel, "Pcie device security enhancements specification," 2018. Available at https://www.intel.com/content/www/us/en/io/pci-express/pcie-device-security-enhancements-spec.html.

[91] C. Moreno and S. Fischmeister, "Accurate measurement of small execution times—getting around measurement errors," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 17–20, 2017.