SIDE CHANNEL ATTACK WITH HAMMING WEIGHT LEAKAGE

by

Mingyang Xia

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2019

Dedication

I dedicate this thesis to my beloved parents, Bin and Yuying. May their lives be full of love, health, and happiness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Information security and data encryption attract more and more attention due to the prevalent use of communication networks and internet of things (IoT). To understand the principles and methods of the information attacks is indispensable to develop more efficient and more secure protection against malicious attacks.

This thesis focuses on the performance of Correlation Power Analysis (CPA) attack, which is one of the most popular methods of Side Channel Attack (SDA). First, we introduce Advanced Encryption Standard (AES) and perform CPA attack using the information leakage of AES to attack the cryptographic devices. Secondly, we evaluate the performance of CPA attack and show data distribution of recorded power consumption for various Hamming Weight across multiple devices. Thirdly, we calculate the best matched distribution of the recorded power consumption and show how the performance of CPA attack changes under different channel (noise free channel and AWGN channel).

LIST OF ABBREVIATIONS USED

AES	Advanced Encryption Standard
AWGN	Additive White Gaussian Noise
ASIC	Application-specific integrated circuit
CMOS	Complementary Metal Oxide Semiconductor
CPA	Correlation Power Analysis
COV	Co-variance
DES	Data Encryption Standard
DPA	Differential Power Analysis
ESF	Electronic Sentinel Foundation
FIPS	Federal Data Processing Standard
FPGA	Field－Programmable Gate Array
HD	Hamming Distance
HW	Hamming Weight
NIST	National Institute of Standard and Technology
NSA	National Security Agency
NMOS	N-Metal-Oxide-Semiconductor
PMOS	Positive channel Metal Oxide Semiconductor
STD-DEV	Standard Deviation
SPA	Simple Power Analysis

ACKNOWLEDGEMENTS

First, I sincerely thank my two supervisors, DrS. Colin O' Flynn and Zhizhang (David) Chen, who introduce me the topic of the side channel attacks. Dr.Colin O' Flynn guided me technically though my course of the study. I have had a really good time working in the professional environment he has created. It has been my honor to be one of his Master students.

I also want to thank my committee members, Dr. Guy Kember and Dr. Jean-Franscois Bousquet, for their reviews of my thesis and their constructive comments and feedback.

I would also like to thank Dr. Liu Zhu and Xiaoyao Feng, who have also worked in Dr. Chen's laboratory, for their valuable discussions and suggestions during our projects. I am very grateful for their ideas and instructions.

I want to say thank the ECED Administrative Secretary, Nicole Smith and Rebecca Baccardax, for their friendly help and patience in administrating of my program.

Finally, I want to thank my parents, Bin and Yuying, for their consistent support and love.

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

With the rapid development and popularization of cryptography, encryption algorithm and corresponding attack methods have become increasingly diverse. Side channel attack is an attack method that first appeared in the mid-1990s, being well developed by Paul Kocher using statistical methods [1]. It is different from the classical cryptographic algorithm analysis technique that used mathematical theory such as algebra and probability to reveal the secret key. It obtains secret information by testing the physical characteristics of cryptographic devices to achieve the purpose of attack. Based on the types of the media, side channel attacks are divided into multiple categories: cache side-channel attacks [2], timing side-channel attack [3], power analysis attack [4], electromagnetic attack [5], acoustic cryptanalysis [6], data remanence attack [7], differential fault analysis attack [8], etc.

The emergence of the attacks has broken people's traditional concept of information security, making the physical security of cryptographic algorithms and their hardware products more of concerns. The relationship between the cryptographic algorithm and side channel attack is like sword and shield. In order to prevent the attacks, more efficient and more complicated cryptographic algorithms are developed. Among hundreds of encryption methods, Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two of the most popular and widely used encryption methods in the world. DES is a symmetric encryption block cipher algorithm. It was considered as Federal Data Processing Standard (FIPS) by US Federal Bureau of Standards in 1976 and was widely employed internationally since then. It is based on a

symmetric algorithm using a 56-bit key. The applications of DES were somewhat controversial because of short length of secret key, secret design elements and the suspicious backdoor. As a result, DES has no longer been used as a secure encryption method, mainly because its 56-bit key is too short. In January 1999, “distributed.net” teamed up with Electronic Sentinel Foundation (ESF) to publicly crack a DES encryption key within 22 hours and 15 minutes. In order to provide the security required for practical use, 3DES was proposed for encryption, although 3DES is also vulnerable to attacks theoretically. In 2001, DES was eventually replaced by a new standard, Advanced Encryption Standard (AES).

Advanced Encryption Standard (AES), also known as the Rijndael encryption in cryptography, is a block encryption standard adopted by the US federal government. AES was published by the National Institute of Standards and Technology (NIST) on FIPS PUB 197 on November 26, 2001[9] and came to effect on May 26, 2002. Since 2006, AES has become one of the most popular algorithms of symmetric key encryption. This standard has been analyzed extensively and is now widely used around the world. Since then, AES has been analyzed extensively and is now widely used around the world.

To crack a cryptographic device using AES encryption algorithm, the most widely used attacking method is Power Analysis Attack which records the leaked power consumption of the device and perform attack. Many power analysis attack methods have been developed; they include simple power analysis [14], differential power analysis [12] and correlation power analysis [10] [11].

This thesis focuses on power analysis attack against AES for it is the simplest and most widely use attack method. We first introduce the principle of AES encryption algorithm

along with the introduction of three attacking methods of power analysis attack. We then move to correlation power analysis (CPA) and evaluate the performance of CPA across multiple cryptographic devices under both noise free channel and additive Gaussian white noise (AWGN) channel. More specifically, Hamming Weight model that is used in CPA will be investigated for its capability and performance.

1.2 Objective

The primary goal of this thesis is to understand how the performance of Hamming Weight model works in the Correlation Power Analysis attack across multiple devices based on the software platform and hardware platforms (Chipwhisperer), the two platforms built and developed by NewAE Technology Inc.

The specific objectives are:

1) Compare the attacking efficiency between DPA and CPA attack
2) In-depth explore the leakage point and leakage information in the CPA attack across multiple devices.
3) Calculate the best distribution model and how closely the data match with Gaussian distribution and its statistical properties (e.g., mean, std-dev, shape) across these multiple devices.
4) Investigate the performance of CPA with HW leakage under the AWGN Channel.

1.3 Thesis Organization

This thesis consists of five chapters. The contents of each chapter are as follow:

Chapter 1 introduces the background and motivation of this thesis as well as the

objectives and the organization of the thesis.

Chapter 2 presents one encryption algorithm (AES) and three major methods of side channel attack: Simple Power Analysis (SPA), Differential Power Analysis (DPA) and Correlation Power Analysis (CPA). Comparisons of them are made in the term of advantages and disadvantages. On this basis, three hypothesis power models of intermediate values are introduced in this chapter along with their respective advantages and disadvantages.

Chapter 3 describes the experimental set ups: software and hardware platforms. It also provides introduction to the devices we aim to perform attack on. It explains how Chipwhisperer work for the experiments that include power measurement, capture and analysis.

Chapter 4 focuses on the performance of CPA using Hamming Weight (HW) model based on Chipwhisperer. We will demonstrate the efficiency of CPA attack in two aspects: correlation coefficient threshold of correct key and wrong key, minimum number of traces capable of successfully detecting the correct key. The second part of the chapter focuses on calculation of the best distribution, degree of matching between the data with Gaussian distribution and its statistical parameters (i.e. mean, std-dev, shape) across these multiple devices. The last part of the chapter evaluates and compares the performance of CPA using HW model based on Chipwhisperer across various voltage and frequency under two different channels: noise free channel and AWGN channel.

Chapter 5 concludes the thesis. Future works and directions in this research field are also presented in this chapter.

CHAPTER 2 THE SIDE CHANNEL ATTACKS

This chapter presents the encryption algorithm of Advanced Encryption Standard (AES) along with some fundamental structures and characteristics of Complementary Metal Oxide Semi-Conductor (CMOS). It also provides the review of side channel attack along with three different hypotheses for the power consumption models. Based on the information, three attack methods of side channel attack are demonstrated: SPA, DPA and CPA.

When it comes to the cryptographic devices, there are several essential components for encryption: hardware, cryptographic software, memory and interface. As for cryptographic hardware and software, the hardware part is responsible for performing all cryptographic operations, the software part is implementing all the cryptographic algorithms and making sure that the implementation is compatible with the hardware. The memory contains all the data and information related to cryptographic operations, such as secret keys, intermediate value, plaintext, and ciphertext. Interface answers for communicating between the devices and outside, such as data input and output, receive special commands from outside and, most importantly, protects the cryptographic key from accessing from the outside.

In the digital circuits, there are two manufacturing methods to mount all those components to one single chip or separate chips. The first of methods is called Application Specific Integrated Circuit (ASIC) which is to be an integrated circuit designed for a specific purpose [15]. ASIC have the advantages of smaller size, lower power consumption, higher reliability, improved performance, enhanced confidentiality, and lower cost compared with general-purpose integrated circuits in mass production [17]. The second one is called Field Programmable Gate Array (FPGA) which is a circuit

design done in the hardware and description language (Verilog or VHDL) can be quickly burned to the FPGA for testing through simple synthesis and layout [16]. FPGA is the mainstream of modern IC design verification technology. These editable components can be used to implement some basic logic gates (such as AND, OR, XOR, NOT) or more complex functions such as decoders or mathematical equations. The advantages of FPGA are that users can change their logic functions at any time, and use them flexibly, without intervening in the layout and routing of the chip.

As shown in Figures 2-1, 2-2, and 2-3, there are some cryptographic devices applied in daily life. No matter what methods the cryptographic devices are designed and manufactured with, they all need logic cells. In the next part, we will introduce the structure and characteristics of logic cells that are widely used in the digital circuits.

Figure 2-1.AES encryption chip [13]

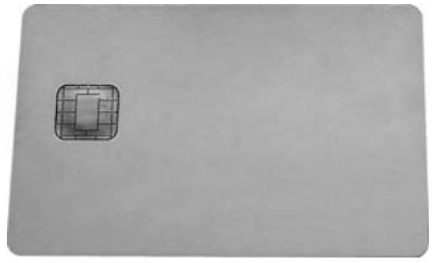


Figure 2-2.Cryptographic smart card [13]

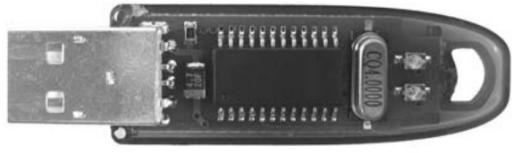


Figure 2-3.Cryptographic smart card in a USB token [13]

2.1 Structure and Characteristic of Digital Circuits

When it comes to digital circuits, power consumption is one of the most essential specifications. Digital circuits draw currents from the power supply and distribute it to every component of the circuits. In this part of chapter, we illustrate the power consumption of cryptographic devices in detail and discuss how power measurement works based on digital circuits of CMOS in practice.

The power consumption we measured mostly depends on the number of logic cells in a circuit, connections between these logic cells and how logic cells are built. It is the result of the designs at four levels: system level, circuit level, cell level (logic cells) and transistor level (semiconductor technology) [13]. Here, we focus on the last two levels and will discuss the first two levels in the next chapter.

In CMOS circuits, static power consumption is mainly caused by leakage current. For a conventional CMOS circuit, there is no DC current at steady state ideally and the static power consumption should be zero; but in the reality the static power consumption of the CMOS circuit is not zero due to the presence of leakage current. The CMOS leakage current mainly included: reverse bias PN junction current and sub-threshold current of MOS transistor. Therefore, static power consumption is mainly composed of these two parts. For deep sub-micron MOS devices, there are also many additional leakage currents caused by secondary effects.

The power consumption of CMOS is mainly caused by charging and discharging the load capacitors. The larger number of switching times is, the more power the circuit consumes. Based on this characteristic, we will introduce three methods of side channel attack along with three different type of power consumption models in the next chapter.

2.2 Cryptographic Algorithm

Advanced Encryption Standard (AES) is a round-based block encryption algorithm, which was first designed by Belgian cryptographers, Joan Daemen and Vincent Rijmen[19]. Based on[19], the fixed block length is 128 bits and the length of secret key can be 128, 192 or 256 bits. During the encryption, the input data (plaintext) will be divided into 16 bytes, each byte contains 8 bits. The AES encryption performs in a 4×4 matrix of bytes. In every encryption round, the round key is generated from the original secret key; there are four ways to do the generation: AddRoundKey, SubBytes, ShiftRows and MixColumns.

AddRoundKey: As shown in Figure 2-4, the XOR between round secret key and each block of plaintext in the matrix is performed. The matrix of round key is the same size as

the byte matrix.

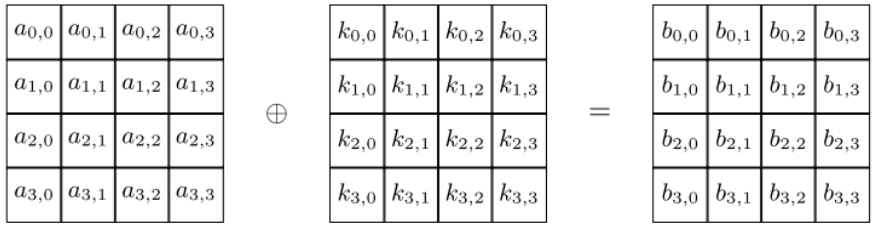


Figure 2-4. AddRoundKey

SubBytes: As shown in Figure 2-5, S-box is developed that contains 128 substitution elements. Each byte in the output of first operation matrix has a corresponding substitution in the S-box. This operation is the only nonlinear transformation among these four operations. This step provides the ability to transform non-linearly in cryptography, which avoids simple algebraic attacks.

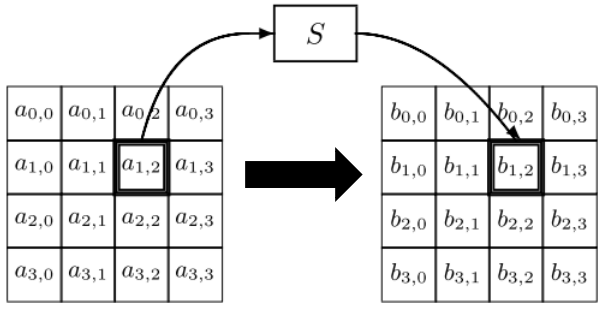


Figure 2-5. SubBytes

ShiftRows: As shown in Figure 2-6, each line of the operation matrix is cyclically shifted to the left by an offset. In AES (block size 128 bits), the first line remains unchanged, and each byte in the second line is looped one space to the left. Similarly, the offsets of the left and fourth rows to the left are 2 and 3 respectively.

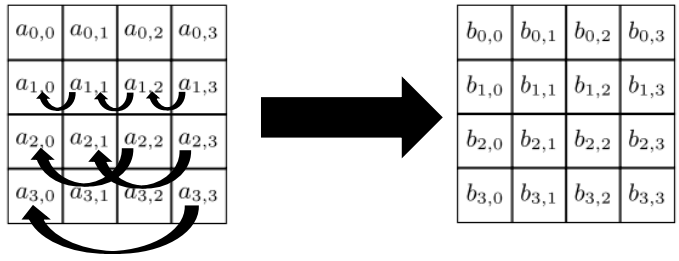


Figure 2-6. ShiftRows

MixColumns: As shown in Figure 2-7, four bytes of each column are combined with each other through a linear transformation. The four elements of each column are treated as 1, x^1, x^2, $x^3$with one polynomial combination. Note that MixColumns is omitted in the last round of encryption; instead AddRoundKey performs again during the last round.

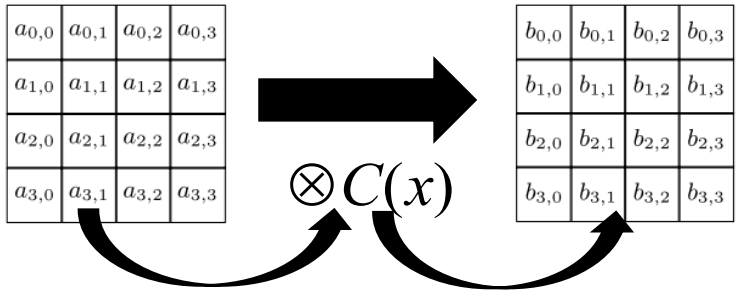


Figure 2-7. MixColumns

During the AES encryption [20], there are normally 10 rounds of encryption. At the beginning, the round key is generated and loaded into a cryptographic device. While the plaintext is loading, the AddRoundKey operation is performed at the same time. After initiating the encryption, four operations of transformation are performed successively in order of SubBytes, ShiftRows, MixColumns and AddRoundKey. This procedure repeats 8 times before the encryption enter the final round. In the last round, the MixColumns operation is bypassed and the intermediate values go directly into the AddRoundKey operations. The flow chart of AES encryption is shown as followed in Figure 2-8:

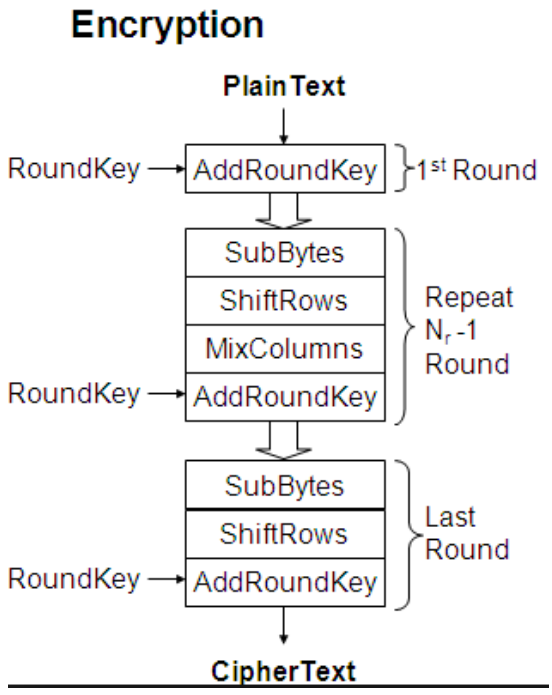


Figure2-8. AES Flow Chart

2.3 Side Channel Attack

Side channel attack [21] (SCA) is a method of attacking an encryption device with the leakage of side channel information such as time consumption, power consumption, or electromagnetic radiation. Among them, power analysis attack is one of the most powerful methods [23], including Simple Power Analysis attacks (SPA), Differential Power Analysis attacks (DPA) and Correlation Power Analysis attack (CPA). The effectiveness of CPA is more significant than the mathematical method of cryptanalysis [22], thus posing a serious threat to cryptographic devices.

In the modern cryptographic devices, various mathematical algorithms of encryption are widely used to protect the confidential information, such as human fingerprint [24], image transmission [25], and cellular message transmission [26]. There is usually an

input text (known as plaintext) and a secret key or key chain. The devices run the plaintext and the secret key under a certain algorithm to get an output text called cipher-text. Generally, the algorithms of cryptographic devices are known by public [27]. However, the secret key is kept in confidential all the time. During the encryption, all the operation between the plain-text and the key is executed by digital circuits such as those with CMOS. As we mentioned before, no matter if the transistor of CMOS is ON or OFF, there will be power consumption generated. Based on the cryptographic algorithms we know and the power consumption of a digital circuit we measure, secret key can be detected and found.

2.3.1 Simple Power Analysis

Simple Power Analysis (SPA) is "a technique that involves directly interpreting power consumption measurements collected during cryptographic operations" [28]. In other words, the attack is trying to find secret keys of cryptographic devices by looking into only one or a small amount of power traces. There are three types of SPA that are mostly used in practical including: visual inspection of power traces [29], template-based SPA [30] and collision attacks [31]. The first of method requires a good knowledge of the mechanism inside the cryptographic devices, and the last two of methods ask for mathematically statistic methods to extract more information in the single power traces.

2.3.1.1 Visual Inspection of Power Traces

The method of visual inspection of power traces, like its literal meaning, is a method that distinguish some of the instructions of cryptographic devices by only inspecting the measured power traces. If the succession of the instructions depends on the secret key, the visual inspection of power traces can lead to the secret key and poses a severe danger

of security.

As we mentioned before, smart cards or any kinds of cryptographic devices are built by transistors in semiconductor. When there is current pass through the transistors from Source to Gate, the power consumption is inevitable. For power consumption of cryptographic devices, some of them are data-dependent, others are operation-dependent. In other words, when the devices encrypt different data with different secret keys or operate different instructions, power consumption will be different. As a result, the changes in power consumption can reveal the secret keys and the devices are vulnerable. For example, the widely used encryption algorithm AES is a round based algorithm which has ten rounds during the encryption. Each round again has four different round transformations: AddRoundKey, SubBytes, ShiftRows and MixColumns. For different operations in encryption, the features shown in power traces will change differently. The visual inspection of power traces can compare the patterns of measured power traces and translates them to corresponding instructions. After getting information of related instructions, the secret key can be easily revealed.

Figure 2-9 presents that the measured power trace of one full round of AES. In one round of AES, typically, it operates first three round transformations on one byte; next it operates MixColumns on every four bytes together, and then the round key is generated in the end. As shown in Figure 2-9 [29], the first phase is between clock cycle 555 and 1800, the second phase is between clock cycle 1800 and 4305, and the third phase is between clock cycle 4305 and 4931. Due to the characteristics of a typical AES implementation, the first phase of measured power traces corresponds to the instructions of AddRoundKey, SubBytes and ShiftRows, operating only on one byte. The second phase corresponds to MixColumns, operating four bytes together; the round key is

generated in the third phase.

As shown in Figure 2-10 [29], we zoom in on the first phase from clock cycle 738 to 958. It is very likely to figure out the sequence of instructions of the cryptographic algorithm by the visual inspection of power traces.

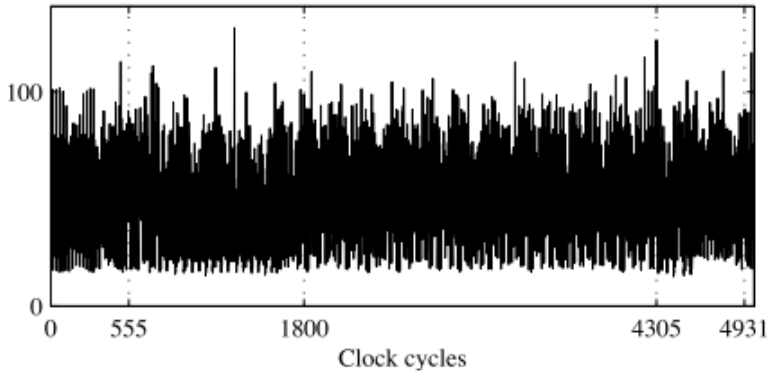


Figure 2-9. One Round of AES (power trace has been compressed) [29]

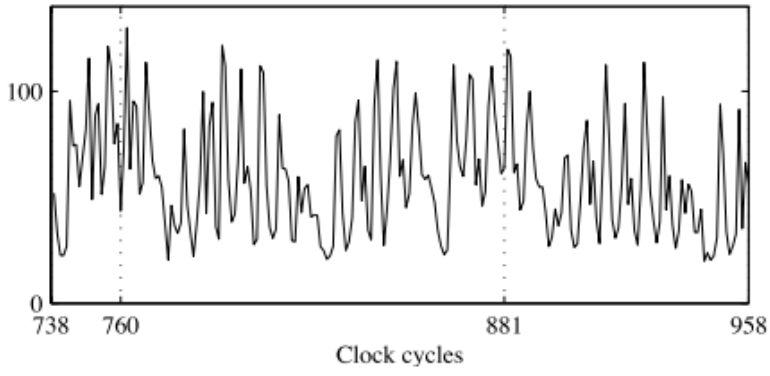


Figure 2-10. Sequence of AddRoundKey, SubBytes and ShiftRows operations [29]

2.3.1.2 Template-based Simple Power Analysis (SPA)

In a template attack, it is assumed that the attack can characterize the characteristics of

the attacked device, which means that the attacker can determine the template of certain instruction sequences. The attack method is to use the templates and the power trace obtained from the attacked device to determine the key. The probability is calculated by taking the recorded power trace into the probability density function of the multivariate normal distribution. Template-based SPA exploits the fact that power consumption depends on the data being processed by cryptographic devices [32]. Unlike other power analysis attacks, template attacks usually consist of two phases. The first phase characterizes the power consumption feature called the Template, and the second phase uses this Template to perform the attack. In other words, template attack is a trained attack.

In the first phase, also called as profiling phase, the attacker chooses another device that has the similar type or algorithm with the device under attack. In contrast of the device under attack, this device is fully under the attacker's control, which means the attacker can manipulate the device to suit their purpose. To get the Template, attacker runs certain sequences of instructions with different data and keys in the selected device and records the corresponding power traces. It is assumed in the template attack that all recorded power traces can be characterized using a multivariate normal distribution defined by the mean vector and the co-variance matrix, called a template.

As for the chosen keys, for example, there are only 256 possible keys in the 8-bit micro-controller. How to wisely choose the data for building the template is essential in the template attack. The first strategy is to conduct exhaustive searching. We build up a template for every pair of data and keys. That means there are 256^2 templates in our hand. To simplify the workload, we use intermediate values of data and keys. For example, we can use suitable functions $f(d_i, k_j)$, such as S-box $S(d_i \oplus k_j)$, to narrow the number

of templates down to 256. Each of 256 templates can be assigned to 256^2 pairs of data and keys. To achieve more efficiency and further simplification, power models is a better choice in the power analysis attack. For example, we can use Hamming Weight (HW) to modify the functions as the second strategy. We build templates by calculating the hamming weight model of the output of S-box. The number of templates decrease to only 9, and each of the template corresponds to a value of Hamming Weight model (0-8). These nine templates can also be assigned to the original 256^2 pairs of data and keys.

In the second phase of template-based attack, known as matching phase, we input data (same as the data used in the profiling device) and key (the secret key of device that is under attack) and then record one corresponding power trace. We calculate the probability between the recorded power traces and the template we build in the profiling phase with the formula below (2.1):

$$p(t;(m,Cov)_{d_i,k_j}) = \frac{\exp(-\frac{1}{2} \bullet (t-m)' \bullet Cov^{-1} \bullet (t-m))}{\sqrt{(2 \bullet \pi)^T \bullet \det(Cov)}} \tag{2.1}$$

The magnitude of the probability value reflects the degree of matching of the given energy trace with the template. Intuitively, the correct template should correspond to the highest probability. Since each template corresponds to a key, information of the correct key can also be given. Based on the maximum-likelihood decision rule, the correct key is corresponding to the highest probability of the template. In other words, the template with correct key fits the recorded power trace best using (2.2).

$$p(t;(m,Cov)_{d_i,k_l}) > p(t;(m,Cov)_{d_i,k_j}) \tag{2.2}$$

The essence of template attack is the classification algorithm, in which the template is

the classifier. Based on the input power consumption, the template is used to identify the power trace as an intermediate value in the encryption calculation process (such as the round key, the Hamming weight of the S-box output, and the mask in the high-order attack.). In the standard template attack, the Gaussian recognition algorithm is used as the classification algorithms.

2.3.2Differential Power Analysis

Differential Power Analysis (DPA), which was first proposed by Paul Kocher and Joshua Jaffe in 1996 [1], is a very popular attack method in recent days. Because of its high efficiency and low cost, it has found different applications, such as digital forensics [33] and decryption of the McEliece Crypt−system [34] and AES implemented FPGA [35]. It has different attack pattern compared to SPA we mentioned before. The essence of SPA attack is to analyze the patterns of power consumption in time to figure out the operations information, even to reveal the secret key. However, in the DPA attack, these time dimension information of power traces are not as important as those in SPA. It focuses more on data dependent information between the recorded power traces while SPA look into the operation information from only one or few numbers of recorded power traces instead. That is why more numbers of power traces are necessary for performing a DPA attack. The advantage of DPA compared to SPA is that it can still reveal the secret key without knowing a lot of details of the device under attack, and even with a lot of noise in the recorded power traces.

As we mentioned before, most of digital circuits of the cryptographic devices consist of different kinds of transistors. The most significant power consumption is caused by switching these transistors between ON and OFF. During the performing encryption algorithm in the cryptographic devices, power consumption depends on the processed

data. Generally, there are three steps in a DPA attack [36]: firstly, an intermediate function of the algorithm is selected as the differential function and the value of the differential function is calculated; secondly, the power consumption of cryptographic devices in encrypting different data is measured; and the data we measured and recorded is analyzed and processed.

Here, we take the AES algorithm as an example to illustrate the attack principle of DPA. As mentioned before, AES is a cryptographic algorithm with 10 encryption rounds. In each round, there is a non-linear transformation called S-box which we choose as our intermediate function of DPA attack. The attacking steps are as followed:

1) Data Classification

In the first step of DPA attack, we choose first bit of S-box output in the first encryption round as the Most Significant Bit (MSB) b. This MSB depends on the candidates of cryptographic key and the input plaintext we choose. During the attack, we calculate the intermediate values with the pairs of all possible key hypotheses $\vec{k}(k_1,k_2,...,k_K)$ and known plaintexts $\vec{d}(d_1,d_2,...,d_D)$; and then we divide the intermediate values into two different categories: $b=0, b=1$.

2) Data Collection

After classification of data, we measure power consumption and record power traces. we focus on the power consumption after first round of encryption. Those corresponding input plaintexts that are involved in the first step $\vec{d}(d_1,d_2,...,d_D)$ are sent into the cryptographic device, while we record power traces $\vec{t}(t_1,t_2,...,t_T)$ which consist of

matrix T of size $D \times T$. Average power consumption of these power traces is calculated as the reference trace t_r. Based on the two different categories in the first step, we record the power traces of the cryptographic key and the plaintext that corresponds to $b = 0$ as $\vec{t^1}(t_1^1, t_2^1, ..., t_T^1)$ and do the calculation to get the average power traces t_{avg}^1. For the part of $b = 1$, the same procedure is performed to get the average power traces as t_{avg}^2.

3) Data analysis

In this step, we carry out the differential calculations between the average power traces t_{avg}^1 and t_{avg}^2 to get a differential curve. From this differential curve, we can tell if the key guess is correct or not. If there are several significant peaks shown in the curve, the key guess has a good possibility to be the correct one; otherwise, the key guess is wrong. When the correct key is used to get the intermediate value, the corresponding power traces will be affected and show the difference between the average power traces of these two categories. On the contrary, if the wrong key is involved in the algorithm, power consumption has no relation with it and there is no significant peak in the corresponding differential curve.

DPAattack has gradually become the most popular attack method in the modern days, mainly because it dependson the data information collection and analysis. This attack technology has good performance and decryption efficiency. It works on the fact that power consumption changes with different operations. These changes can be detected and analyzed using special electronic gauges and mathematical statistical techniques to obtain specific key information. In addiction, this attack method is more robust against the noise than any other encryption methods. The attack does not require any information

about the individual power consumption of each device. Once the attacker knows the output of the algorithm and the corresponding power consumption traces, the attack can be started right away.

The example results of the DPA attack is shown in Figure 2-11to Figure 2-15:

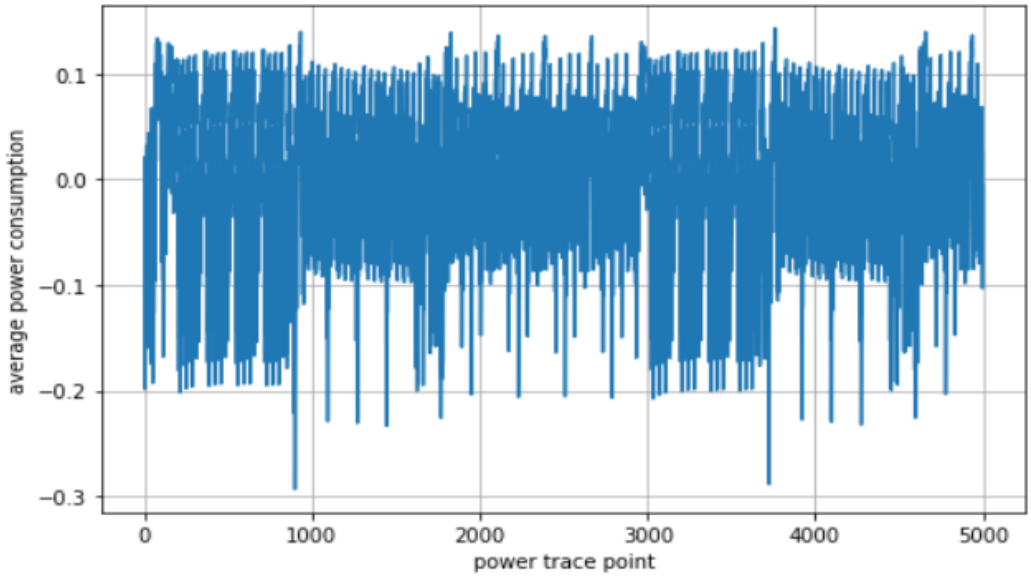


Figure 2-11.Average power consumption

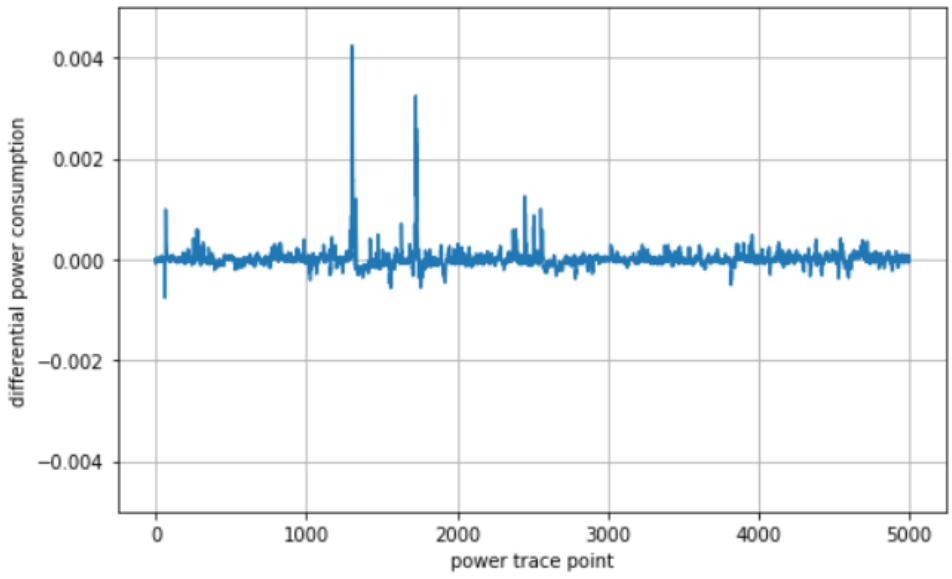


Figure 2-12. Differential power consumption (correct key)

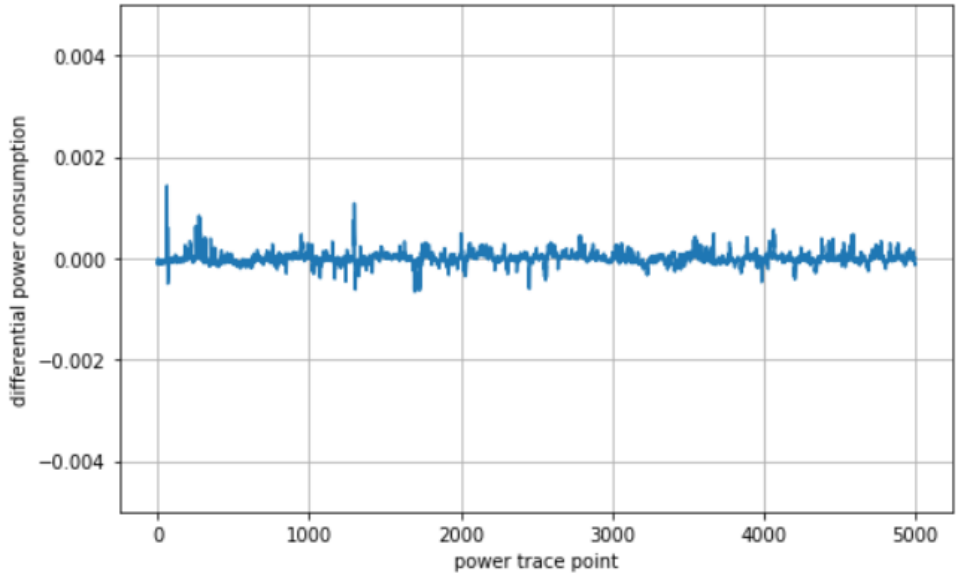


Figure 2-13. Differential power consumption (wrong key 1)

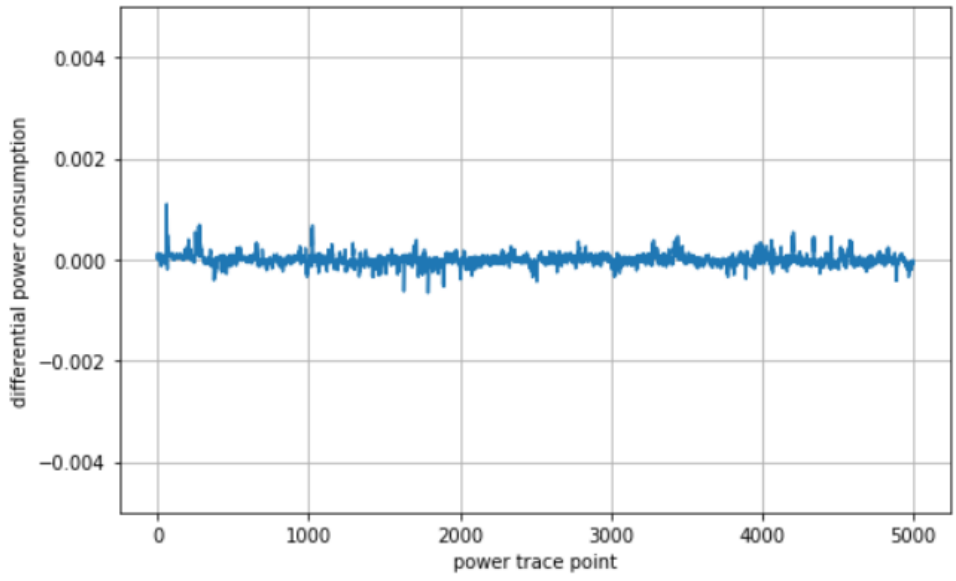


Figure 2-14. Differential power consumption (wrong key 2)

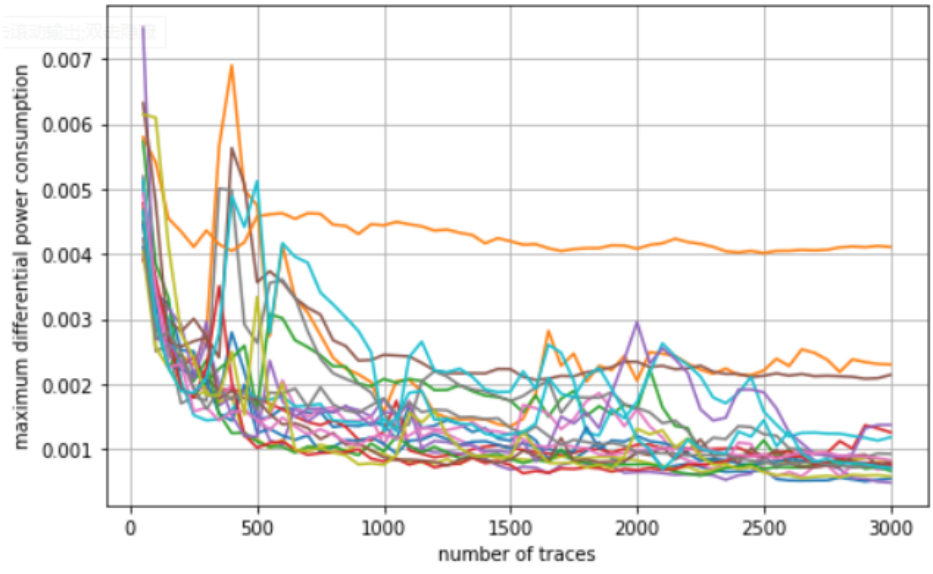


Figure 2-15. Number of traces

As shown in Figure 2-15, with the DPA attack, we choose the first bit of output of first round S-Box as the most significant bit. The differential power consumption is around 0.004 for the correct key, as it is around 0.0025 for the wrong keys. Figure illustrates that

at least 600 power traces need to be recorded to perform the attack and reveal the secret key successfully.

2.3.3Correlation Power Analysis

Compare to the DPA attack, correlation power analysis (CPA) attack exploits a more powerful and accurate binary model to describe measured power consumption. Therefore, to some extent, CPA attack is more powerful than DPA attack and much harder to defend. Correlation Power Analysis (CPA) attack mainly uses the Hamming weight model and calculates the correlation coefficient between the Hamming weight and the corresponding power trace. The larger the correlation coefficient is, the stronger the correlation between them is. The correct key always has the largest correlation coefficient during the CPA attack.

Since byte replacement occupies most of the power consumption, usually only the relationship between the output of the S-box and recorded power traces are considered in the CPA attack. In AES, the key is split into 16 sub-keys, each of which is one byte and involves one round of the cryptographic algorithm. In each of decryption round, the attack can focus on cracking one byte (8 bits) of secret key, which needs to be traversed from 0 to 255. The CPA attack steps are as followed:

1. Choose intermediate result from the executed algorithm

In the first step of CPA attack, we choose an intermediate result from the executed cryptographic algorithm as follows (2.3):

$$f(d,k) \tag{2.3}$$

In this function, *d* is known data which is the data input into the cryptographic device to get the recorded power traces in the fourth step. For this purpose, the attacker needs to fully control the selection of data. In most of attack, this data is either plain-text or cipher-text. The input keys of this intermediate result are a small part of whole secret key.

2. Calculate hypothetical intermediate values with the selected data and guess keys

After choosing the function of intermediate result, we calculate the hypothetical intermediate value with the selected data and the candidate keys. For the data, we have D data blocks (2.4):

$$\vec{d}(d_1,...,d_i,...,d_D) \tag{2.4}$$

For the candidate keys, we map all possible choices into the vector (2.5):

$$\vec{k}(k_1,...,k_j,...,k_K) \tag{2.5}$$

After we calculate every pairs of data and key hypotheses, we get a matrix of intermediate value V of size $D \times K$. The calculation formula is as followed (2.6):

$$V_{i,j} = f(d_i, k_j); i = 1...D, j = 1...K \tag{2.6}$$

Every column is the result of different data and the same key hypotheses k_j, every row is the result of the same datawith all possible candidate keys. Because these candidate keys are only a small part of the secret key, the workload of calculation is not very much. The computer can run all the calculation result in second level.

3. Map hypothetical intermediate values to power consumption values

The intermediate values we get in the second step is in digit sequence. In order to compare these intermediate values with the recorded power traces to get the results, we need to map hypothetical intermediate values $v_{i,j}$ to power consumption values $h_{i,j}$. The choice of different types of power models directly influences the performance of final decision making. There are three most common power models to choose in the practical attack which are Bit model, Hamming Distance Model and Hamming Weight model. We will demonstrate these three different types of power models in detail later. Using one of these power models, every element in the matrix H is corresponding to the elements in the matrix V.

4. Measure the power consumption and record the power traces

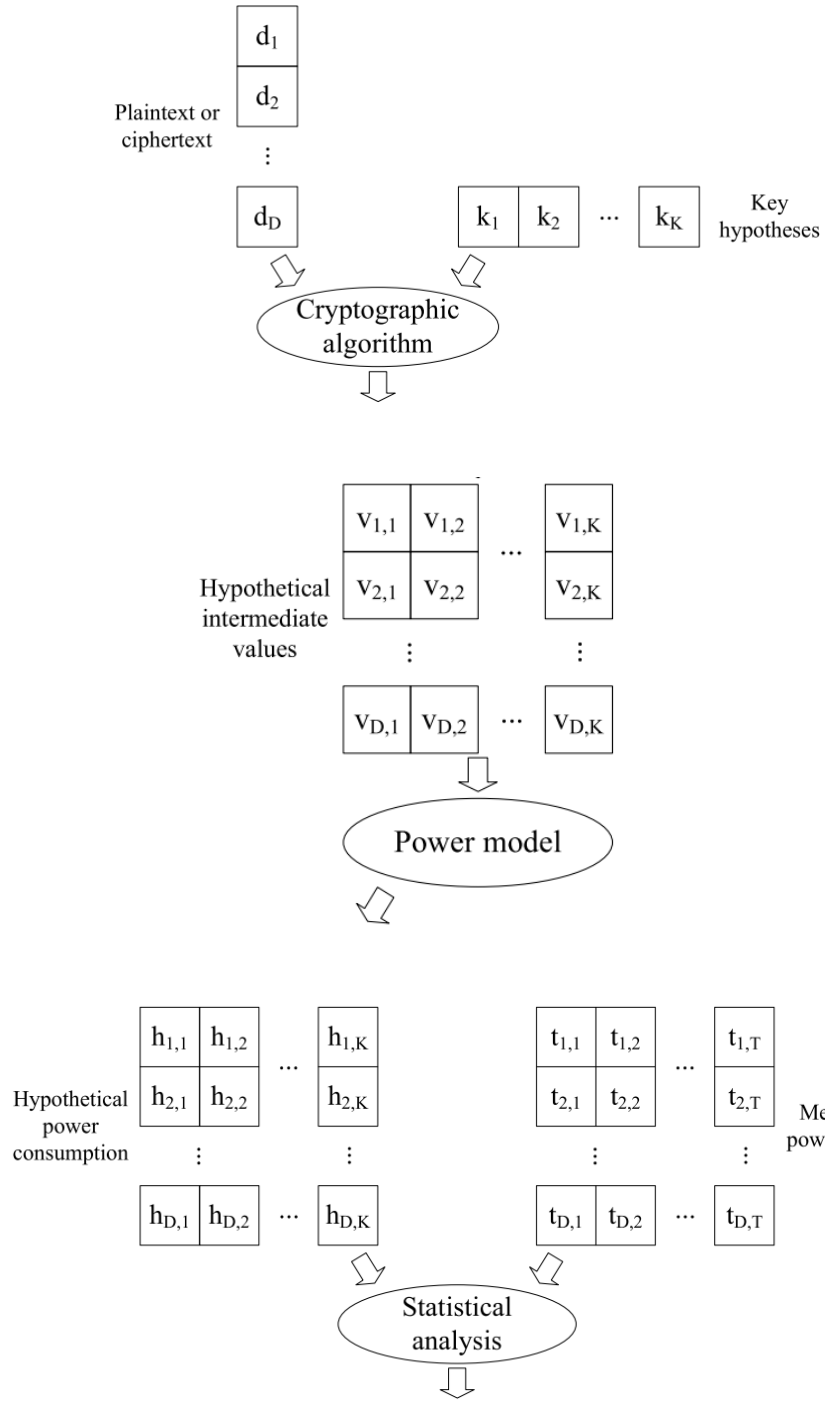
All those first three steps shown above are focusing on calculating the hypothetical intermediate values with the chosen data and candidate keys. In this step, we run the corresponding data, which has already been chosen in step 1, inside the cryptographic device and measure the power consumption during the processing. Corresponding to every data block d_i, we record the power trace as (2.7):

$$\vec{t_i}(t_{i,1},\ldots,t_{i,l},\ldots,t_{i,T}) \tag{2.7}$$

T refers to the total number of sample points in the recorded power traces. There are recorded power traces with sample points in the matrix $T_{i,l}$ of size $D \times T$. Noted that the recorded power traces need to be well aligned, which means that every column in the matrix $T_{i,l}$ is the calculation result of the same operations.

5. Compare hypothetical power consumption values with the recorded power traces

After getting the matrix H of hypothetical power consumption value and the matrix T of recorded power traces, we can finally move to the final step. The attacker compares hypothetical power consumption values of each key hypotheses to the corresponding recorded power traces, which means we do the calculation between each column h_i of matrix H and each column t_j of matrix T. The results of this comparison store in the matrix R of size $K \times T$. Each one of element in the matrix R contains the result of comparison between each column h_i of matrix H and each column t_j of matrix T. These results show how much relative between the hypothetical power consumption value and the recorded power traces. In the CPA attack, the stronger the relation between these two values, the more likely the key hypotheses are correct. That means we make a threshold value to separate the result of correct key to those of wrong keys in the practical attack. In some scenario, all the results in the matrix R are very likely the same. That is because the number of power traces we record is still insufficient. In this method of attack, the number of power traces determines the performance of the attack results. The more power traces we measured, the more precisely the attacker can locate the correct key. Fig. 2-16 shows that the flow chart of CPA attack.


Plaintext or ciphertext
d1
d2
dD
k1
k2
kK
Key hypotheses
Cryptographic algorithm
Hypothetical intermediate values
v1,1
v1,2
v1,K
v2,1
v2,2
v2,K
vD,1
vD,2
vD,K
Power model
Hypothetical power consumption
h1,1
h1,2
h1,K
h2,1
h2,2
h2,K
hD,1
hD,2
hD,K
t1,1
t1,2
t1,T
t2,1
t2,2
t2,T
tD,1
tD,2
tD,T
Measured power traces
Statistical analysis

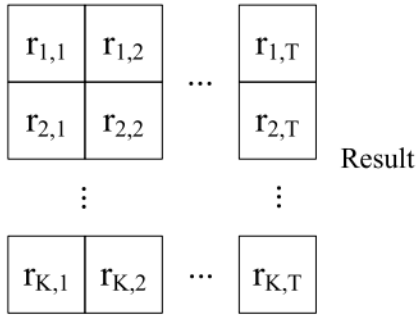


Figure 2-16. Flow Chart of CPA attack [13]

In the final step of CPA attack, the result matrix $R_{T\times K}$ is very essential. To get a better description of the relationship between the hypothetical intermediate values h_i of power consumption and the recorded power traces t_j, correlation coefficient is the first choice. We use correlation coefficient between the columns of $H_{K\times D}$ and the columns of $T_{T\times D}$ to calculate the result $R_{T\times K}$. The formula is shown as followed (2.8):

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i}) \bullet (t_{d,j} - \bar{t_j})}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i})^2 \bullet \sum_{d=1}^{D}(t_{d,j} - \bar{t_j})^2}} \tag{2.8}$$

The values of $\bar{h_i}$ and $\bar{t_j}$ denote the mean values of columns h_i and t_j. Here, we present some of the result and the performance of CPA attacking AES encryption as the example, shown in Figure 2-17, 2-18, 2-19. We compare these three figures to Figure 2-13, 2-14, 2-15. It is obvious that DPA attack needs at least 500 recorded power traces to reveal the secret key successfully, instead, CPA attack only need to obtain a couple of power traces (around 20 traces) to achieve a successful attack. In addition, the correlation co-efficiency of the correct key is up to 0.85 in CPA attack, compared to 0.004 in the

DPA attack. Obviously, the efficiency of CPA attack is much higher than the DPA attack with less recorded power traces. Therefore, in the next chapter, we will focus on the performance of CPA attack.

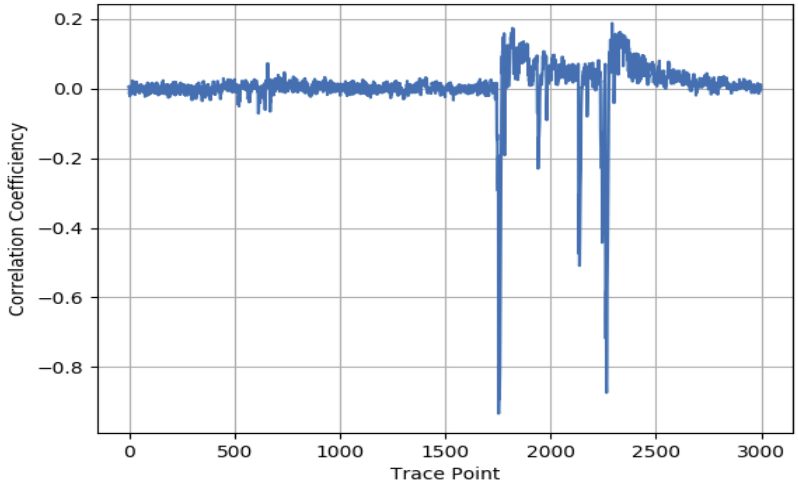


Figure 2-17. CPA Attack (Correct Key)

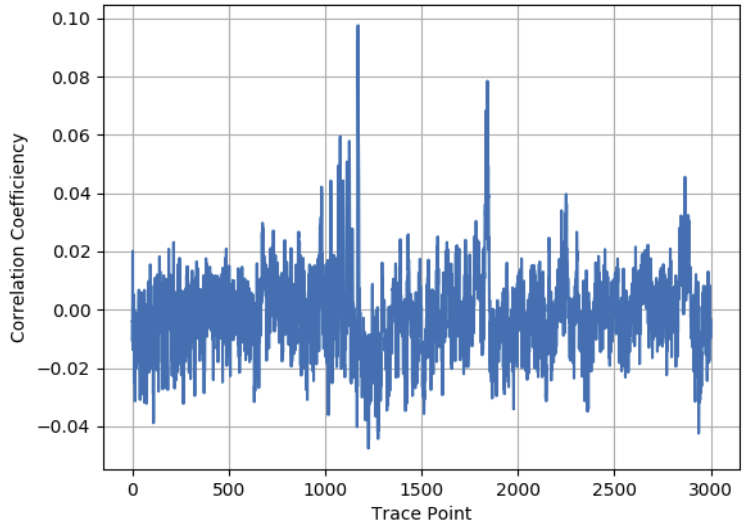


Figure 2-18. CPA Attack (Wrong Key)

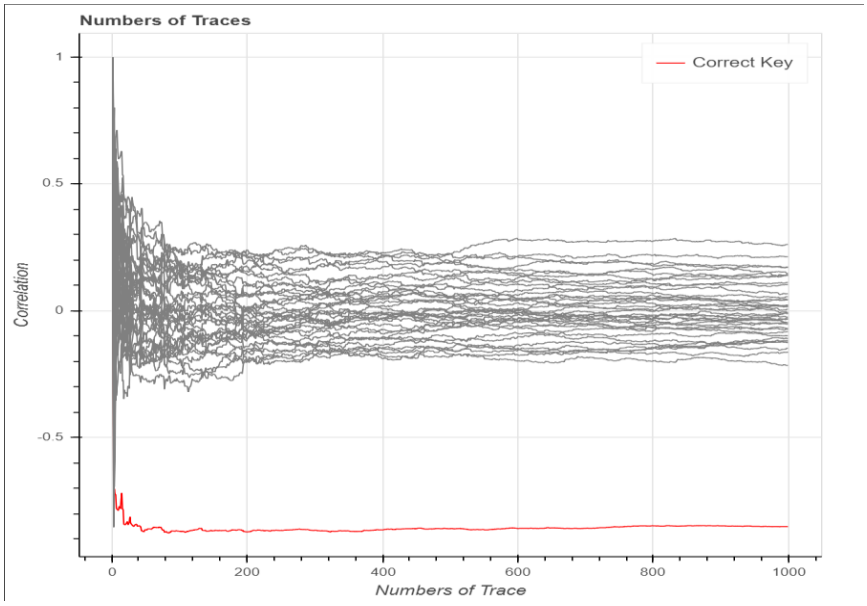


Figure 2-19. Number of Power Traces

As shown in Table 2-1, these three attacking methods use different power consumption models based on different working principles. CPA has the highest attacking efficiency among them.

Table 2-1. Comparison between SPA, DPA and CPA

	Required number of traces	Attacking efficiency	Power consumption model	Working Principle
SPA	One or a few	low	/	PT depends on the operations
DPA	many	high	Bit Model	Differential PC
CPA	many	highest	HW/HD	Correlation between PC and IV

(PC: Power Consumption; IV: Intermediate Value)

2.4Hypothesis Model of Intermediate Value

During the CPA attack, the power model is a crucial factor that decides how successful the attack efficiency would be. The more precisely the description of hypothetical intermediate values is, the higher possibility of successful attack will be.

2.4.1 Hamming Distance Model

Hamming distance (HD) is named after Richard Wesley Hamming, first introduced and used in the Information Theory [37]. In information theory, the Hamming distance between two equal-length strings is the number of different characters corresponding to the positions of the two strings. In other words, it is the number of characters that need to be replaced by converting one string to another. Usually, we perform the operation of XOR between every corresponding position of two strings and counting the number of 1s to get the hamming distance. We refer the intermediate value as D and the reference state value as R .The calculation formula of HD Model is as followed (2.9):

$$W = HD(D,R) = a \cdot HW(D \oplus R) + b \tag{2.9}$$

Where W is the hypothetical power consumption, a present the gain and b is the noise.

In order to perform Hamming Distance Model during the CPA attack, the attacker need to know more detail of the cryptographic device, such as the state of a cell before and after the operations [38]. We take AES implementation as an example, in the last round of encryption, the input data of S-box is stored in the same register as the output of S-box. There is information leakage that can be described by HD model. In the last encryption

round, the XOR is performed between the last round secret key and an intermediate value, followed by the operation of SubBytes and the operation of ShiftRows. As we mentioned before, in the operation of SubBytes, 128 bits of input data are divided into 16 bytes (every byte contains 8 bits). Each of 16 bytes has a corresponding substitution as the output of S-box. Therefore, we choose the ciphertext of the last round as the reference state value and the input data of S-box as the intermediate value in the last round. Calculate the HD between these two chosen values as the hypothetical power consumption and get the correlation coefficient between the hypothetical power consumption and the measured power traces. The block diagram of CPA using HD model is as followed in Figure 2-20:

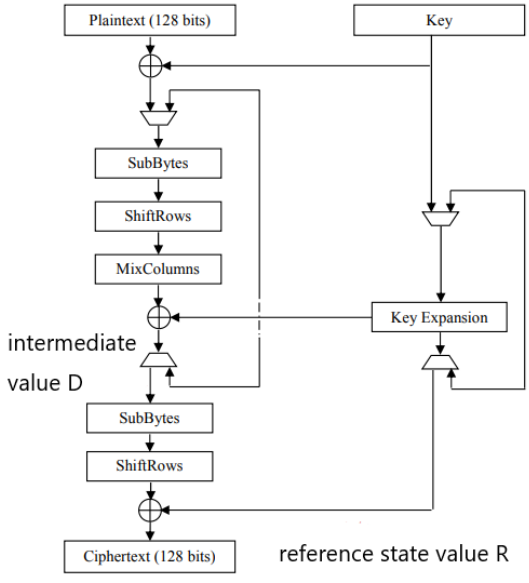


Figure 2-20. Block Diagram of CPA using HD model

2.4.2 Hamming Weight Model

Hamming Weight Model is a special case of Hamming Distance Model where the reference state is zero. When performing the CPA attack using HW model on the

cryptographic devices, we only need to count the number of 1 in the intermediate value[39]. The calculation formula of HW model is shown as followed (2.10):

$$W = a \cdot HW(D) + b \tag{2.10}$$

In this case, the attacking procedure of CPA attack using HW model is much less complicated than the one using HD model. Therefore, the attacker is required to know less details of the cryptographic devices than before. In the case of AES implementation, we aim at the output of S-box in the first round of encryption as the intermediate value. Figure 2-21 to 2-23 illustrate the performance of CPA attack using HW model.

CHAPTER 3 EXPERIMENT PLATFORM

In this chapter, we introduce three encryption chips (CW308T-XMEGA, STM32F071, STM32F303) that are under attack as a target hardware and Chipwhisperer-lite as capture hardware. As for software part, we use Virtual Box as a platform to run Chipwhisperer Jupyter on.

3.1 Hardware

3.1.1 Capture Hardware (Chipwhisperer-Lite)

The Chipwhisperer-Lite board [41] is used for capturing the power consumption of the target devices and perform various kinds of attack (Timing Analysis Attack, Clock Glitch Attack, Voltage Glitch Attack, Fault Attack, DPA Attack, CPA Attack etc.) based on these recorded data. It mainly consists of one SPARTAN-6 FPGA, one Atmel ARM based 32-bit flash micro-controller and one 20 pin connectors connected with the target board.

SPARTAN-6 FPGA Product Table is as followed:

Table 3-1.SPARTAN-6 FPGA Product Table [45]

Part Number	XC6SLX9
Slices	1430
Logic Cells	9152
CLB Flip-Flops	11440
MaxDistributed RAM (Kb)	90
Block RAM (18Kb each)	32

Part Number	XC6SLX9
Total Block RAM (Kb)	576
Clock Mgmt Tiles	2
MaxSingle-Ended I/O Pins	200
MaxDifferential I/O Pairs	100
Memory Controller Blocks	2

Microchip's ARM®-based SAM3U2C [44] is a member of the SAM3U family of flash micro-controllers based on the high-performance 32-bit ARM Cortex®-M3 RISC processor. The Parametric Table is as followed:

Table 3-2.Microchip's ARM®-based SAM3U2C Parametric Table [44]

PART NUMBER	ATSAM3U2C
MAX.CPU SPEED (MHZ)	96
PROGRAM MEMORY SIZE (KB)	128
SRAM (KB)	36
PART NUMBER	ATSAM3U2C
TEMPERATURE RANGE (C)	-40 TO 85
PART NUMBER	ATSAM3U2C
OPERATING VOLTAGE RANGE (V)	1.62 TO 3.6
NUMBER OF USB MODULES	1
ADC INPUT	8
MAX.ADC SAMPLING RATE (KSPS)	1000
MAX.ADC RESOLUTION (BITS)	12
INPUT CAPTURE	6
MAX.I/O PINS	57

INTERNAL OSCILLATOR	4,8,12 MHZ, 32KHZ

The 20-pin connector is designed for communication between the Chipwhisperer main board and the target device board. The pinout is as followed:

Table 3-3. 20 Pin-connector

Pin Number	Name	Dir	Description
1	+VUSB(5V)	O	Not Connected on Chipwhisperer-Lite
2	GND	O	System GND.
3	+3.3V	O	+3.3V to Target Device
4	FPGA-HS1	I/O	High Speed Input (normally clock in).
5	PROG-RESET	I/O	Target RESET Pin (AVR Programmer).
6	FPGA-HS2	I/O	High Speed Output (normally clock or glitch out).
7	PROG-MISO	I/O	SPI input: MISO (for SPI + AVR Programmer).
8	V-Target	I	Desired I/O voltage in range 1.5V-5V.
9	PROG-MOSI	I/O	SPI output: MOSI (for SPI + AVR Programmer).
10	FPGA-TARG1	I/O	Target IO Pin 1 - Usually UART TX or RX.
11	PROG-SCK	I/O	SPI output: SCK (for SPI + AVR Programmer).
12	FPGA-TARG2	I/O	Target IO Pin 2 - Usually UART RX or TX.
13	PROG-PDIC	I/O	PDI Programming Clock (XMEGA Programmer), or CS pin (SPI).
14	FPGA-TARG3	I/O	Target IO Pin 3 - Usually bidirectional IO for smartcard.
15	PROG-PDID	I/O	PDI Programming Data (XMEGA Programmer).

Pin Number	Name	Dir	Description
16	FPGA-TARG4	I/O	Target IO Pin 4 - Usually trigger input.
17	GND	O	
18	+3.3V	O	
19	GND	O	
20	+VUSB(5V)	O	Not Connected on Chipwhisperer-Lite

3.1.2 Target Hardware

3.1.2.1 XMEGA [46]

The main parts of XMEGA Target Board consist of 20 pin connector, SMA glitch connector and SMA measure connector. As shown in Figure 3-1, J2 is a 20-pin connector, which uses the standard NewAE 20-pin connector pinout, connected with the I/O lines of Chipwhisperer-Lite Board. SMA glitch connector, which connected to the SMA measure connector of Chipwhisperer-Lite by default, is used to generate VCC glitch into the VCC pin. SMA measure connector of XMEGA Target Board, which connected to the SMA glitch connector of Chipwhisperer-Lite by default, is used to perform the power consumption measurement.

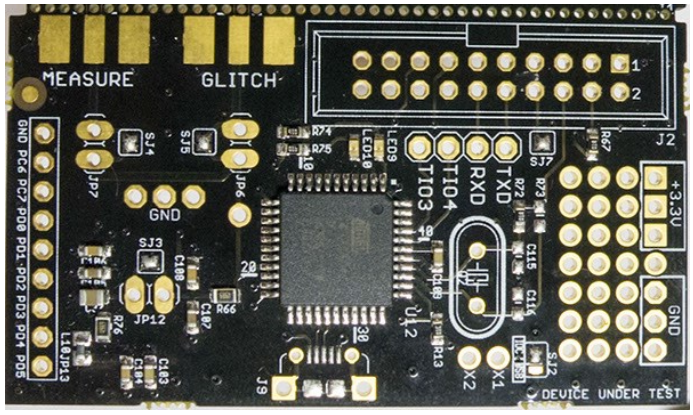


Figure 3-1. XMEGA Target Board

To build firmware and set up the communication with Chipwhisperer-Lite, we run the python script of Chipwhisperer scope. The code is as followed:

```
#connect to chipwhisperer

import chipwhisperer as cw

scope = cw.scope()
target = cw.target(scope)
```

Figure 3-2.Connect to Chipwhisperer

```
# setup scope parameters
scope.gain.gain = 45 # 45
scope.adc.samples = 3000
scope.adc.offset = 0
scope.adc.basic_mode = "rising_edge"
scope.clock.clkgen_freq = 7300000 #7370000  7000000-7800000
scope.clock.adc_src = "clkgen_x4"
scope.trigger.triggers = "tio4"
scope.io.tio1 = "serial_rx"
scope.io.tio2 = "serial_tx"
scope.io.hs2 = "clkgen"
```

Figure 3-3.Scope Parameters Setup

```
from chipwhisperer.capture.api.programmers import XMEGAProgrammer
def program_target(scope, fw_path):
    programmer = XMEGAProgrammer()
    programmer.scope = scope
    programmer._logging = None
    programmer.find()
    programmer.erase()
    programmer.program(fw_path, memtype="flash", verify=True)
    programmer.close()
```

Figure 3-4.XMEGA Programmer Setup

```
program_target(scope, fw_path)
fw_path = "../../hardware/victims/firmware/simpleserial-aes/simpleserial-aes-xmega.hex"
```

Figure 3-5.Connect to XMEGA Target

3.1.2.2 STM32F071 [41]and STM32F303 [42]

The STM32F board supports several STM32F devices in the TQFP-64 package. In the family of STM32F chips, there are F0, F1, F2, F3 and F4. In this thesis, we focus on F0 and F3 as the target.

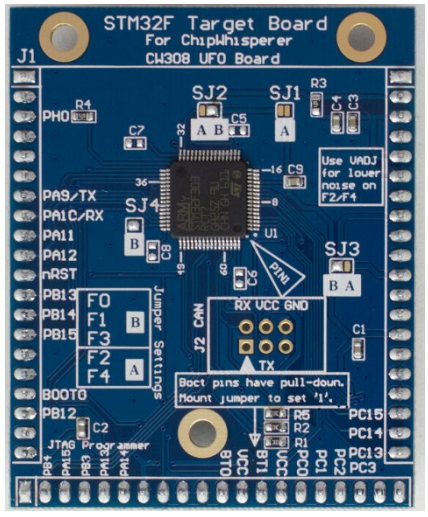


Figure 3-6. STM32F Target Board

The Parametric Table of STM32F0 and STMF32F3 is as followed:

Table 3-4. STM32F Series

STM32F Series	Package	Device	Flash	SRAM	Tested
F0	TQFP-64	STM32F071RBT6	128KB	16KB	Yes
F3	TQFP-64	STM32F303RCT7	256KB	40KB	Yes

The Chipwhisperer connecting setup and Scope parameters setup are as same as XMGEA. STM32F programmer setup and connecting setup are as followed in Figure 3-7, 3-8:

```
from chipwhisperer.capture.api.programmers import STM32FProgrammer
def program_target(scope, fw_path):
    programmer = STM32FProgrammer()
    programmer.scope = scope
    programmer._logging = None
    programmer.small_blocks = True
    programmer.open()
    programmer.find()
    programmer.erase()
    programmer.program(fw_path, memtype="flash", verify=True)
    programmer.close()
```

Figure 3-7.STM32F Programmer Setup

```
program_target(scope, fw_path)
fw_path = "../../hardware/victims/firmware/simpleserial-aes/simpleserial-aes-cwlitearm.hex"
```

Figure 3-8.Connect to STM32F Target

3.2 Virtual Box Platform (Chipwhisperer Jupyter)

Oracle VirtualBox, a virtual machine software, which designed by a Germany company called InnoTek Software, is now being developed by Oracle. It provides users with other virtual x86 operating systems on 32-bit and 64-bit Windows, Solaris, and Linux operating systems. Users can install and execute Solaris, Windows, DOS, Linux, OS/2 Warp, OpenBSD and FreeBSD systems as virtual client systems on VirtualBox. The version of Virtual Box we use is 5.2.30, released in May 13th, 2019. In order to run virtual machine of Chipwhisperer Jupyter on it, we need to install an extension package as well. The general parameters are shown in Table 3-5.

Table 3-5. Virtual Box (Chipwhisperer Jupyter)

Name	Chipwhisperer Jupyter
Type	Linux
version	Ubuntu (64-bit)
Base memory	2048MB
ipv4 address	192.168.33.11
ipv4 network mask	255.255.255.0
DHCP server	Disable

CHAPTER 4 CORRELATION POWER ANALYSIS WITH HAMMING WEIGHT ACROSS MULTIPLE DEVICES

In this chapter, we focus on the performance of correlation power analysis (CPA) attack using Hamming Weight across multiple encryption devices. We first perform the CPA attack on three different devices: STM32F071, STM32F303 and XMEGA. We analyze the recorded power consumption based on two aspects: correlation coefficient threshold of correct key and wrong key, minimum number of traces which can successfully reveal the correct key. Second part of the chapter focuses on calculation of the best distribution of the recorded power consumption, how closely the data match with Gaussian distribution and how changes in the distribution (mean, std-dev, shape) across these multiple devices. Final part of the chapter evaluates and compares the performance of CPA using HW model based on Chipwhisperer across various voltage under two different channels: noise free channel and AWGN channel.

4.1 CPA performance across multiple devices

In order to perform Correlation Power Analysis Attack on the AES, the very first step is trying to record the power consumption traces that correspond to the input data blocks and secret key blocks. Normally, with the AES encryption, for each of these encryption runs, the system will accept 16 bytes of input data known as the plain text and encrypt them with 16 bytes of secret round key generated from the main secret key in each encryption round. We want to reveal how good the AES is and whether it is possible to use Simple Power Analysis to decrypt the secret devices of AES. Figure 4-1 and Table 4-1 shows that we input 1000 pieces of random input data and only one fixed secret key

to the encryption device.

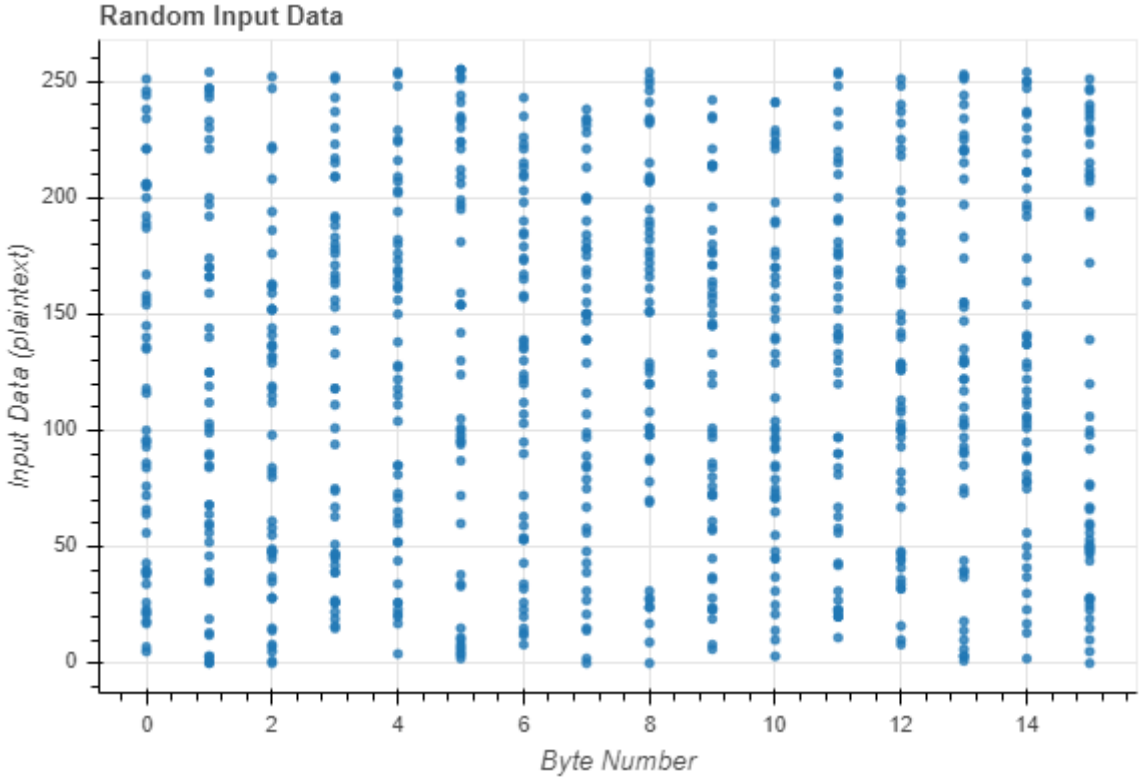


Figure 4-1. Plaintext

Table 4-1. Secret key

Hexadecimal number	bytearray(b'+~\x15\x16(\xae\xd2\xa6\xab\xf7\x15\x88\t\xcfO<')
Decimal number	[43 126 21 22 40 174 210 166 171 247 21 136 9 207 79 60]

As shown in Figure 4-1, plaintext consists of 16 bytes from byte 0 to byte 15. Each byte contains 8 bits which converts to decimal number is 0 – 255. As for the secret key, as we

mentioned before, the main secret key has 16 bytes at the initial phase. During each encryption round, the round key is generated from the main secret key and perform AddRoundKey operation with corresponding plaintext.

With these data and key, we run the Chipwhisperer on it and record the last round of power consumption traces. We randomly pick out four of them and show as in Figure 4-2.

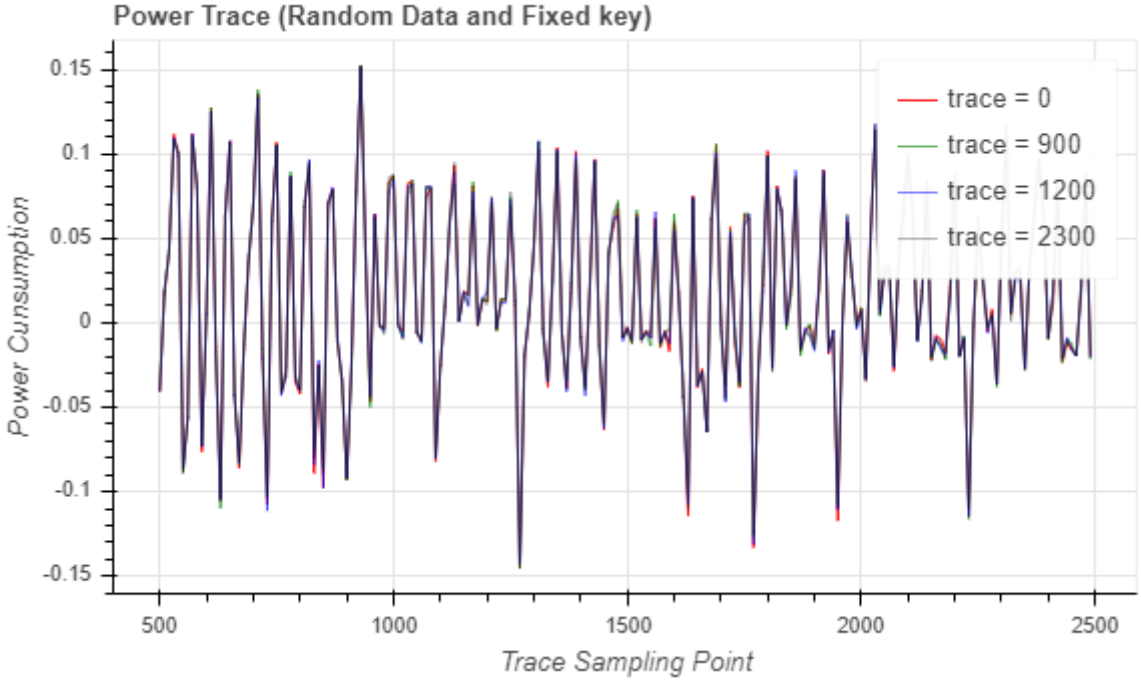


Figure 4-2 Power Trace (Point 500 - 2500)

Figure 4-2 presents that, with the random input plaintext and a fixed key, all the traces are very similar with each other. The power consumption we recorded from the chip is between around -0.15 to +0.15 Volt. Compared them to each other, the traces we recorded follow with the similar pattern. Some slight fluctuations appear during the encryption. The operations of cryptographic algorithm and the feature of S-Box are all

the same based on the AES encryption. These randomly picked four power traces have slightly difference between each other due to the different input data.

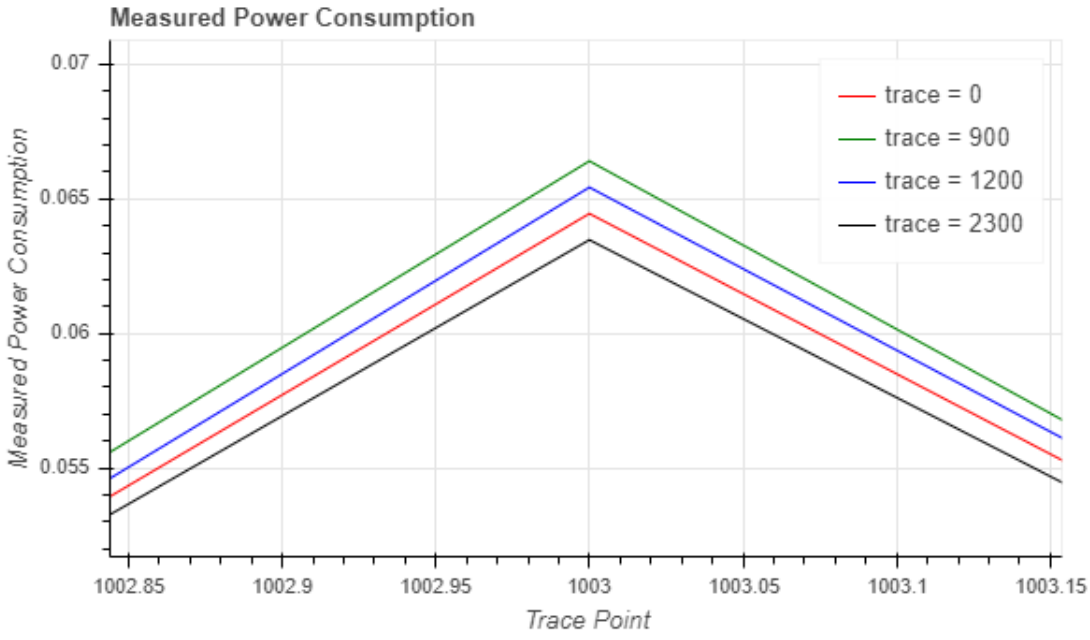


Figure 4-3 Power traces zoom in at point 1003

We zoom in at trace point 1003 to make it clearer in Figure 4-3. The main patterns of these four power traces are similar with each other. Because of the random input plain text, the number of 1 and 0 processed inside the devices is not the same which lead to the different power consumption. The difference of power consumption leaks the information of encryption algorithm. The larger the difference between each power traces, the more information leakage we could get, the better chance we can reveal the secret key. This is the reason that we can use Hamming Weight of intermediate value to analysis and attack the system in the first place.

We only pick out around 200 sample points evenly between trace point 500 and trace point 2500 to draw the plot in Figure 4-2. Now we try to get more points involved and

look at them in a larger picture to see if we can reveal something useful for the decryption from the pattern of the power traces. According to Figure 4-4, 4-5 and 4-6, the trace seems have several different phases. Each one of these phases can be told by the naked eyes as every phase has a certain distinct and discernible pattern. Because of the masking and hiding, all the recorded traces are very alike to each other, only have slight difference caused by the HW information leakage and inevitable noise. However, it is still impossible for the attackers to reveal every bytes of keys in this AES system by just looking at the single trace. On the other hand, as we have already proofed that, with random input of plaintext, the traces we record does not have too much distinct pattern from each other. That is another big reason why it is extremely hard and unstable for the attacker to decrypt the secret devices using Simple Power Analysis.

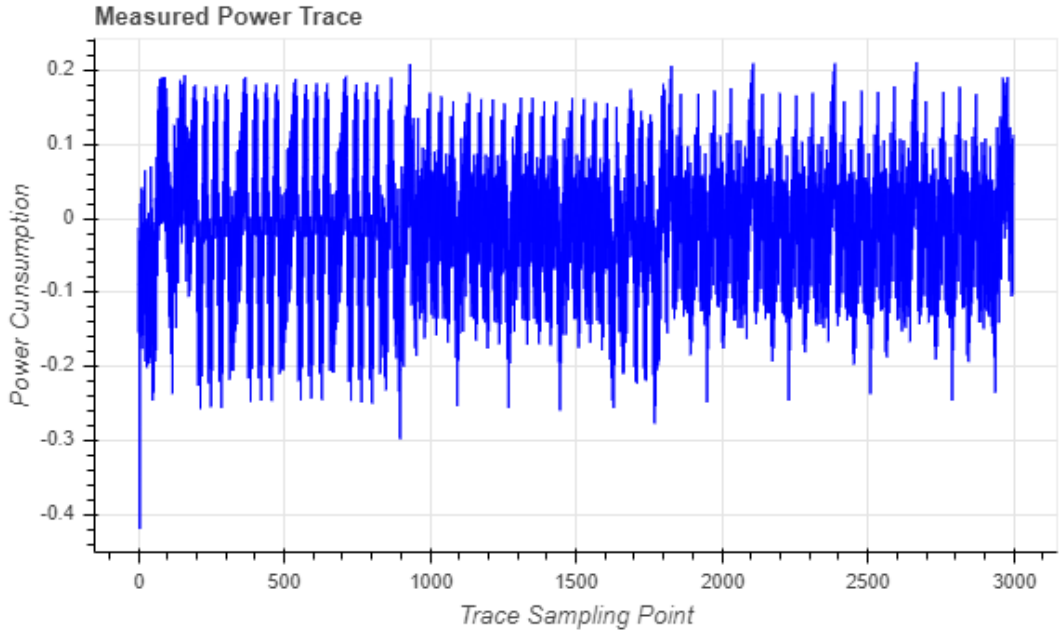


Figure 4-4. Measured Power Traces of STM32F303

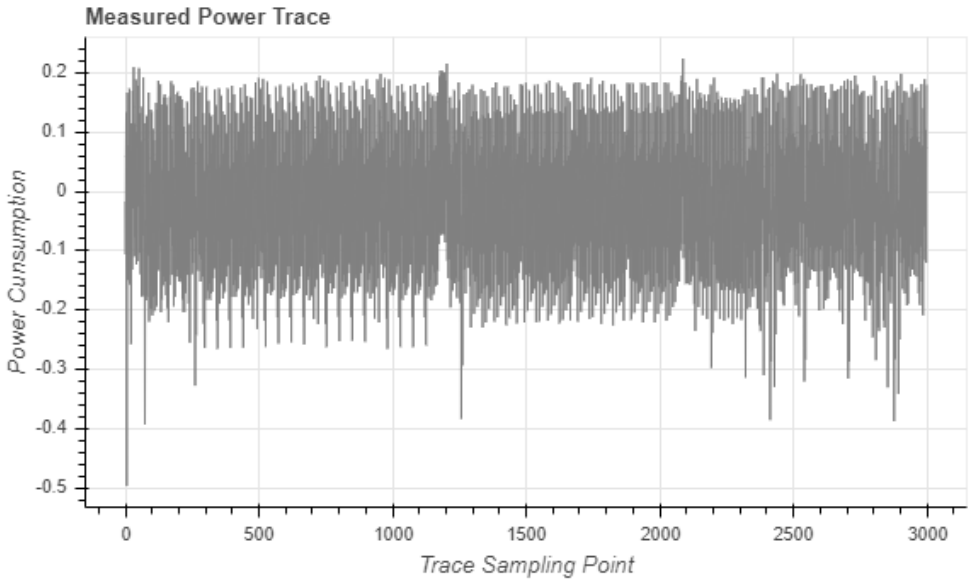


Figure 4-5. Measured Power Traces of STM32F071

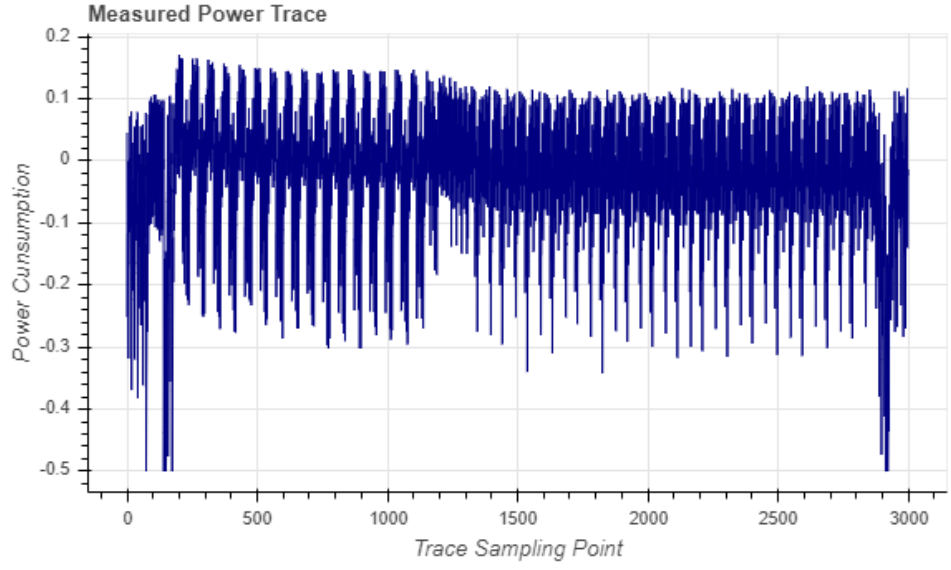


Figure 4-6. Measured Power Traces of XMEGA

We compare the recorded trace plot of STM32F071 with the one of STM32F303 and XMEGA, the power consumption of these three devices are all fluctuating between -0.4to 0.2. The power consumption of STM32F303 is slightly higher than the power

consumption of STM32F071, which may cause by the deviation of CMOS in the circuit. Another interesting thing is that, despite all the recorded traces are similar in the same devices, the power traces are very different across these three chips. As we mentioned before, the encryption method these chips use is AES which is a round-base symmetric block cipher including four major transformations: AddRoundKey, SubBytes, ShiftRows and MixColumns. Among these four transformations, AddRoundKey, ShiftRows and MixColumns are all linear steps. Only SubBytes is nonlinear transformation which is using a built-in matrix called S-box to do the substitution of every single bytes. The first three transformations are all linear and the S-box of these three chips are the same, so the only factor can cause the different power traces between these chips is the circuit structure. Therefore, one thing that the attackers need to know is whether it is possible to attack different devices with the same attacking method and the same capture device and also whether the information leakage of HW is followed in a certain pattern we can exploit during the attack. The main contribution of this thesis is to conclude a general law and principle of Correlation Power Analysis Attack based on decryption of Chipwhisperer across multiple different devices which is using AES.

As we mentioned in the Chapter 2, CPA is more powerful and accurate than the Simple Power Analysis and DPA attack when it comes to reveal the secret keys of the encryption devices. Based on the Hamming Weight Information Leakage, the attacker could find the correct key which has the highest correlation coefficient between the hypothetical key guessing and Hamming Weight Model. Unlike the Simple Power Analysis, CPA attack and DPA attack focuses on using mathematical method to reveal the secret key. As mentioned before, generally, there are three different power consumption models (Bit Model, Hamming Weight Model, Hamming Distance Model) that commonly used in CPA attack and DPA attack. During the experiments here, we are focusing on how the

Hamming Weight Model works across multiple devices based on Chipwhisperer.

From what we have in Figure 4-7 to Figure 4-12, a small subset of the full capture (present 200 sample trace points in each plot) is plotted only. What we illustrate here is plot each of the different "classes" in a different color. With this different brewer method, we should see if there are some interesting spots that have relatively obvious feature in information leakage of Hamming weight. HW is obtained by operating XOR between the input data (plaintext) and the guess key to get an intermediate value and substituting it with the corresponding element in the S-Box. The number of one in that data stream is the HW that we are looking for. When we plot the Figure 4-7, 4-9 and 4-11, the spots that contain more information leakage of HW relate to the darkest red color. Intuitively speaking, at the peak of these traces, the color is much darker than the bottom of the traces. We can more easily figure out which points should be by using the CPA attack which provides more information about where the leakage is happening.

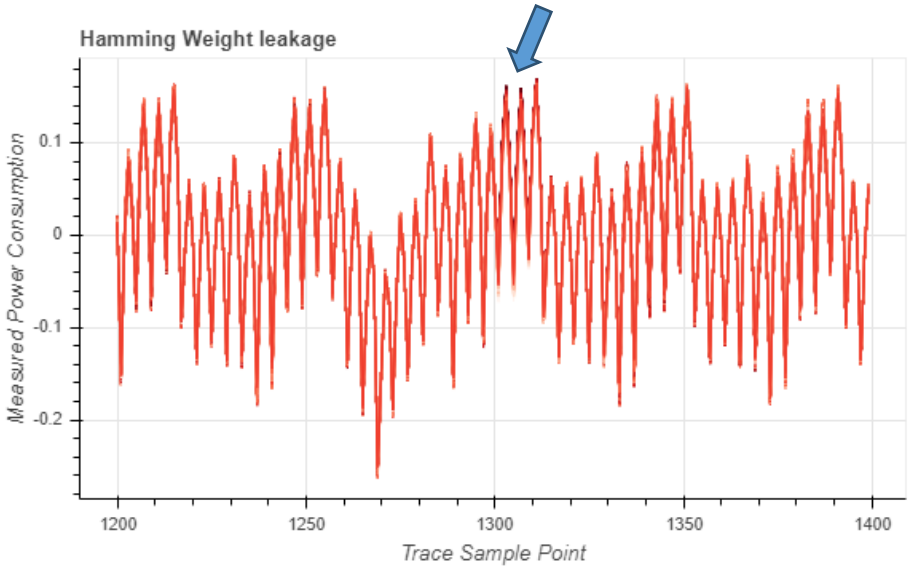


Figure 4-7. Hamming Weight Leakage of STM32F303

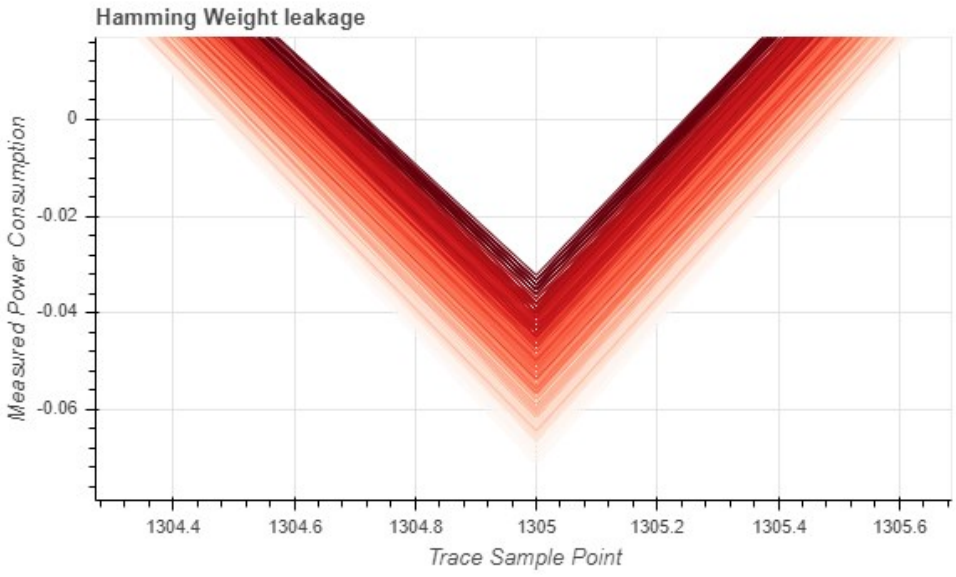


Figure 4-8.Zoom in at point 1305

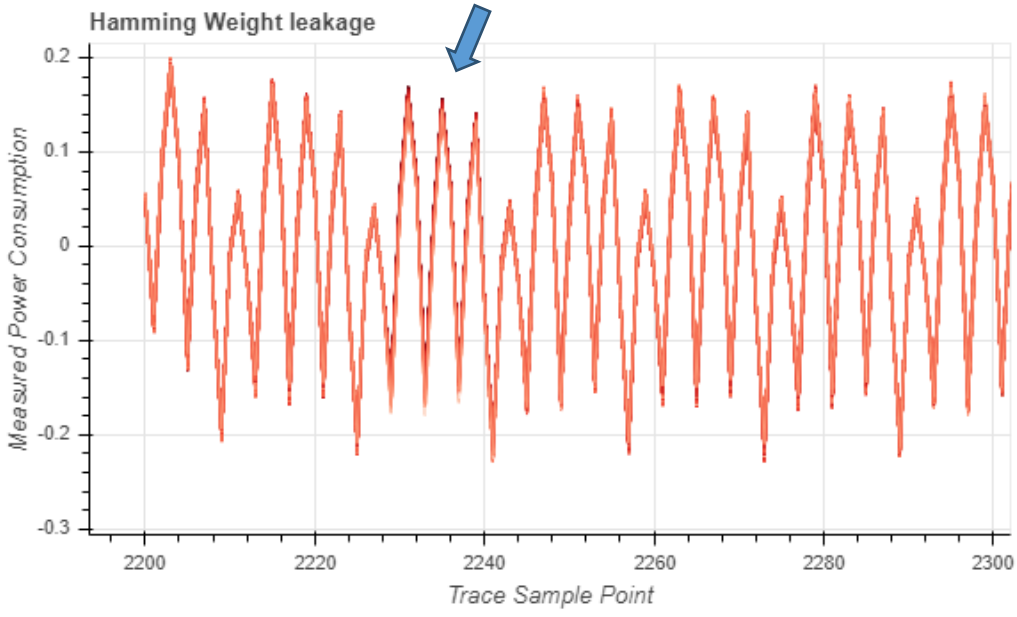


Figure 4-9. Hamming Weight Leakage of STM32F071

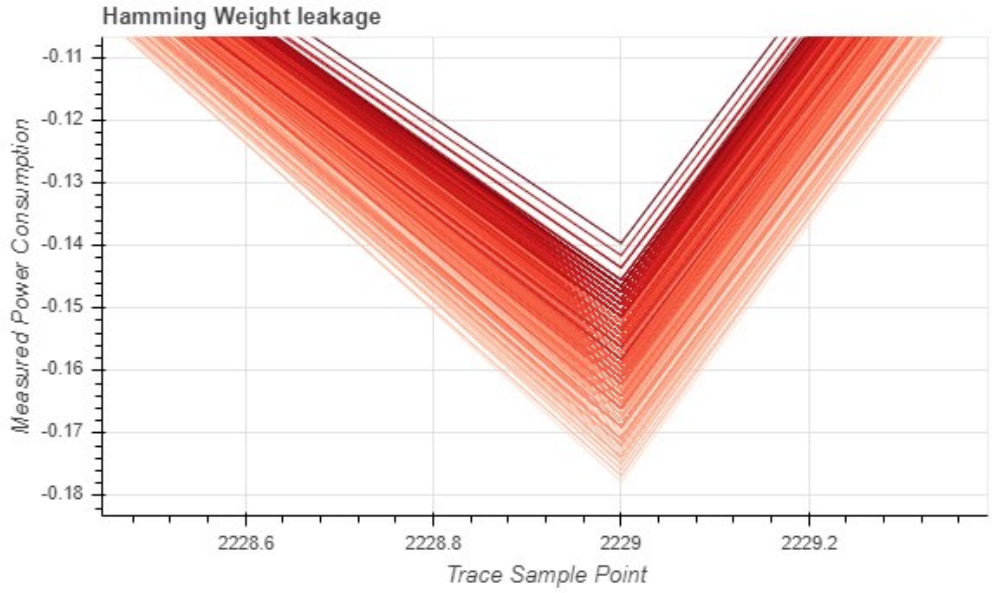


Figure 4-10.Zoom in at Point 2229

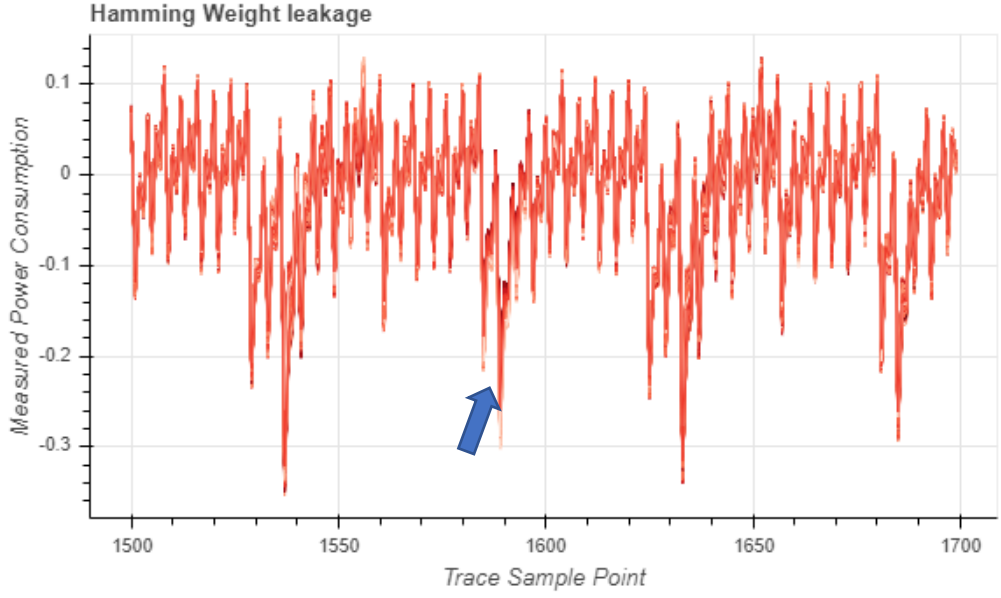


Figure 4-11. Hamming Weight Leakage of XMEGA

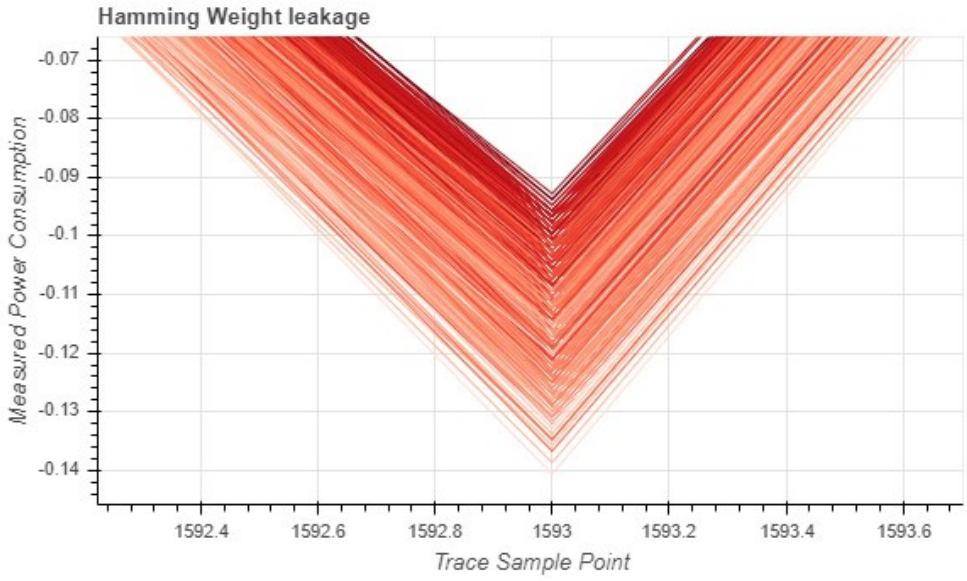


Figure 4-12.Zoom in at point 1593

As shown in Figure 4-8, 4-10 and 4-12, we respectively select trace point 1305 of STM32F303, trace point 2229 of STM32F071 and trace point 1593 of XMEGA, zoom in to see them in detail. At all these three points, the change of color density increases or decreases successively as the power consumption changes. That means the power consumption is related to the HW value of corresponding power traces and the relationship between these two is linear. In other words, the correlation coefficient between the HW value and the power consumption is highest at these spots. As we find these interesting spots, we can reveal the secret key easily.

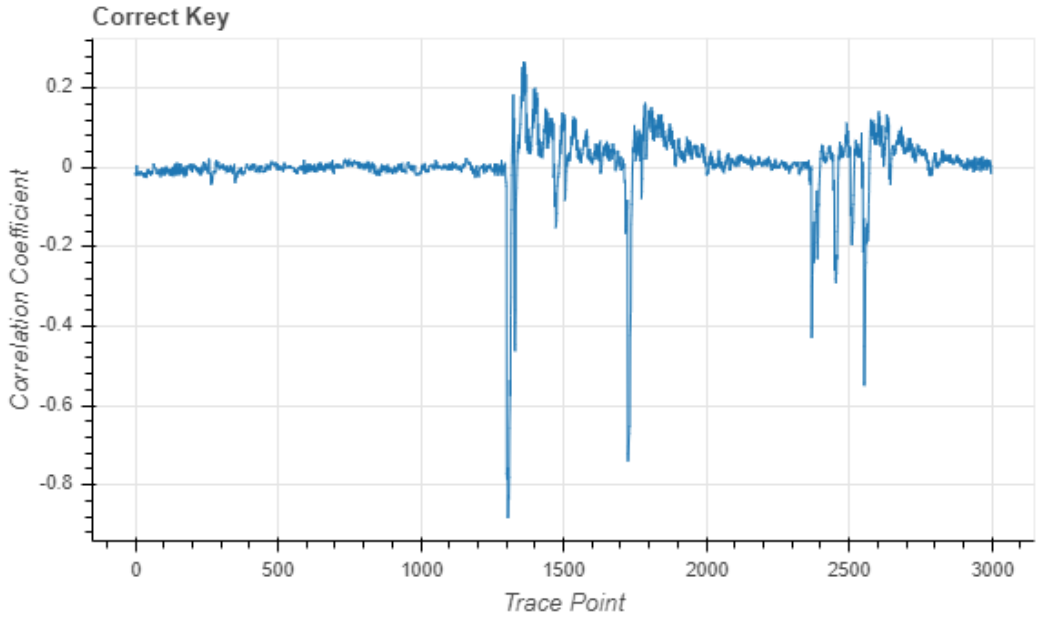


Figure 4-13.STM32F303 Correct Key

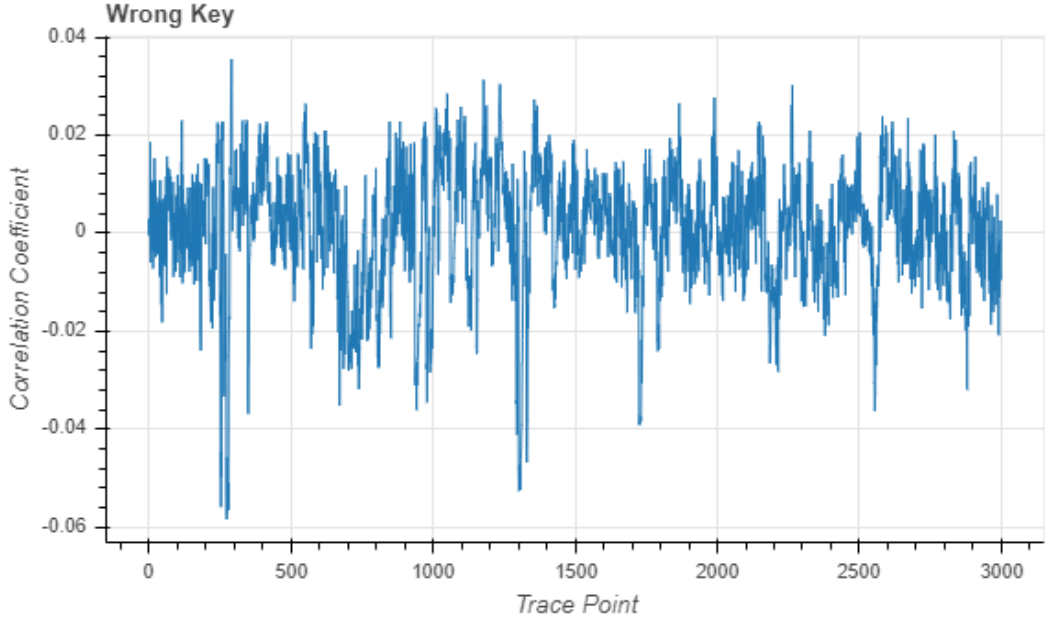


Figure 4-14.STM32F303 Wrong Key

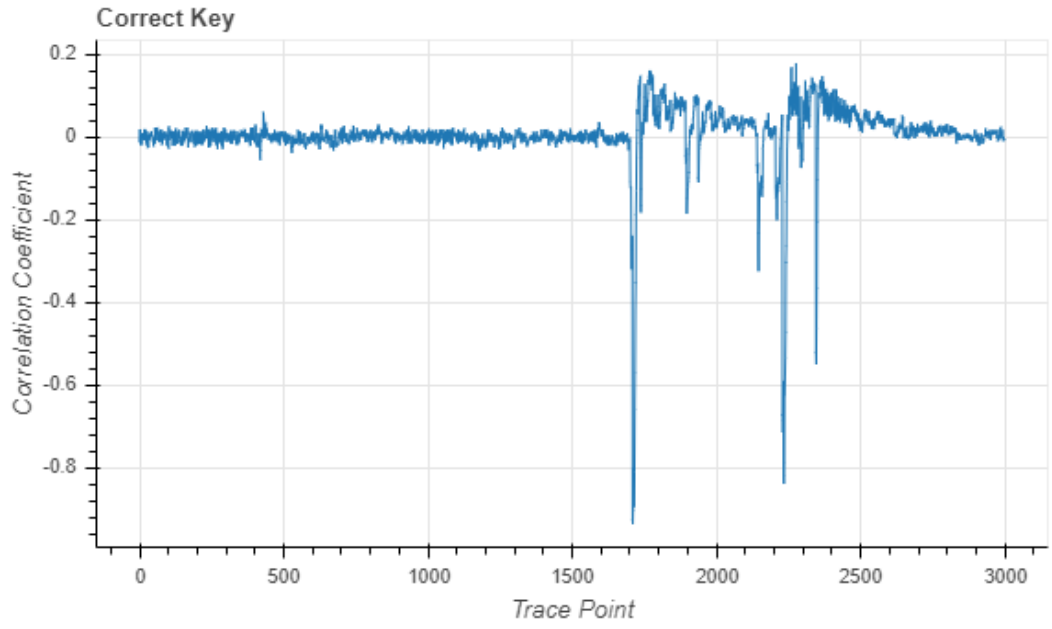


Figure 4-15.STM32F071 Correct Key

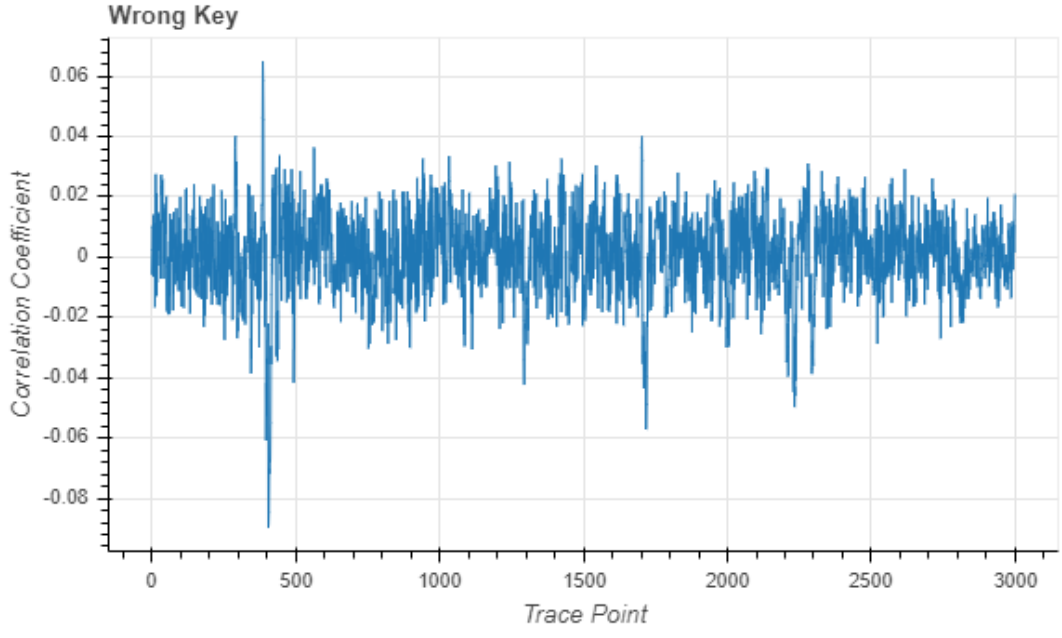


Figure 4-16.STM32F071 Wrong Key

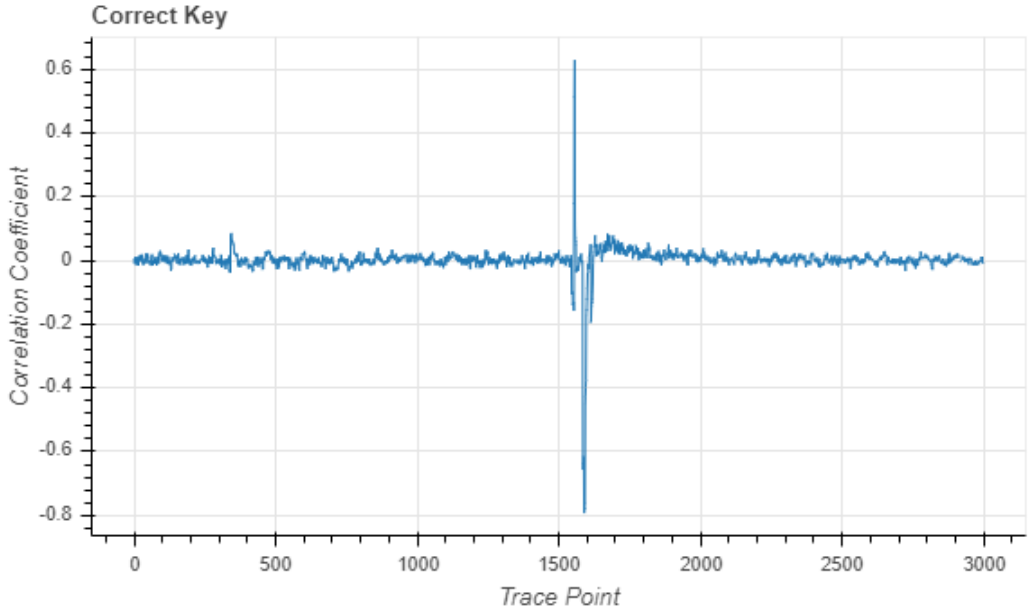


Figure 4-17. XMEGA Correct Key

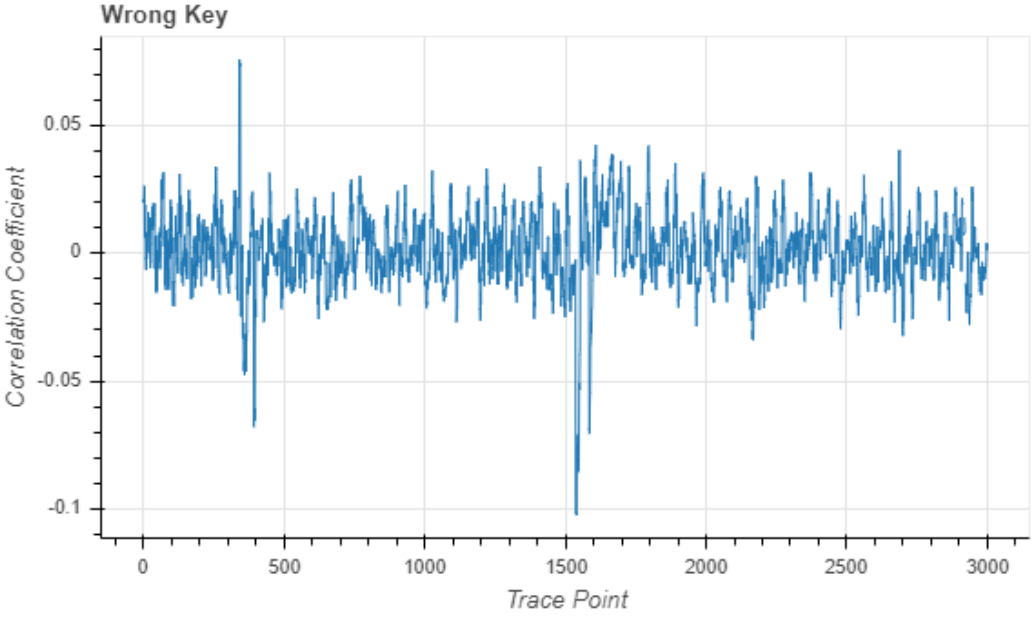


Figure 4-18. XMEGA Wrong Key

To operate a successful power analysis attack using correlation power analysis, the main

goal is to find a leakage point and get the correlation coefficient of the correct key that is distinct from the other ones of the wrong keys. As shown in Figure 4-13, with the correct key guessing, the correlation coefficient of the correct key is almost reach to around 0.85 at the trace point 1305. Compared to Figure 4-13, the correlation coefficient of the wrong key shown in Figure 4-14 is very small. They all fluctuate around 0.04 -- -0.05 which is much smaller than the correct key guessing.

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i}-\bar{h}_i).(t_{d,i}-\bar{t}_j)}{\sqrt{\sum_{d=1}^{D}(h_{d,i}-\bar{h}_i)^2.\sum_{d=1}^{D}(t_{d,i}-\bar{t}_j)^2}} \quad (4.1)$$

As a calculation result (correlation coefficient) of Hypothetical intermediate value of Power Consumption and Measured Power Traces, the higher the value of r is, the stronger and better the matrix of H and T match with each other. So that the correct key can be revealed by observing this result. Set a reasonable threshold, whenever there is a result beyond this threshold, the attacker can secure the correct key right away. The correlation coefficient of correct key and wrong key of STM32F071 and XMEGA are shown in Figure 4-15, 4-16, 4-17 and 4-18, respectively.

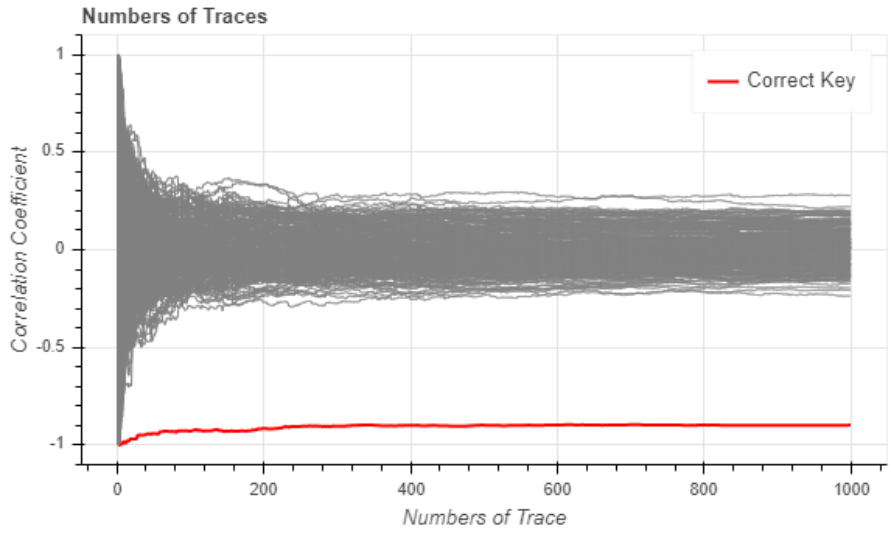


Figure 4-19.Number of Traces of STM32F303

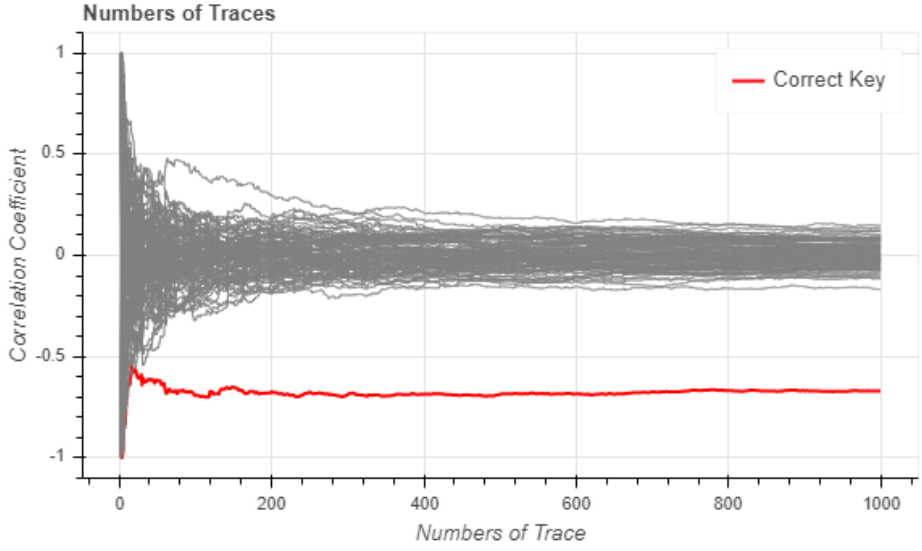


Figure 4-20. Number of Traces of STM32F071

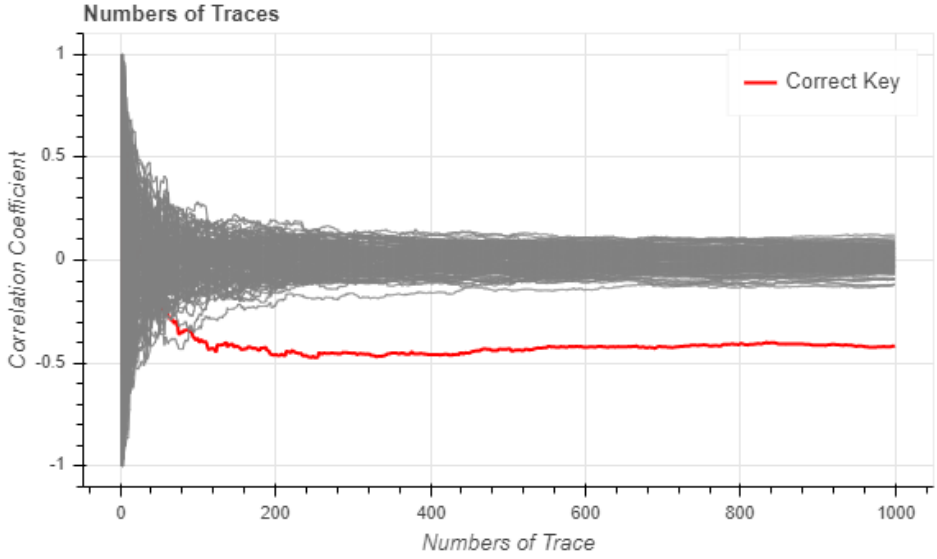


Figure 4-21. Number of Traces of XMEGA

After performing the CPA attack on decrypt AES, every secret key guess is corresponding to one correlation coefficient. The main purpose of this attack method is to distinguish the correlation coefficient between the correct key guess and the incorrect key guess. Also, what we expect is using the recorded traces as few as possible to reveal the correct key.

As shown in Figure 4-19, 4-20 and 4-21, we retry the same correlation coefficient calculation with different trace number to figure out how many traces we need to reveal the correct secret key at least. Clearly, as for STM32F303, the correlation coefficient of the correct key guess distinct with the one of the wrong keys guessing immediately after the number of recorded trace reach to around 10. As for STM32F071 and XMEGA, the performance of CPA attack is not as good as STM32F303. The number of needed traces is 40 and 80, respectively.

4.2 Hamming Weight Model across multiple devices

After we get the correct key base on the correlation coefficient observation, the next interesting thing that we focus on is how the performance of Hamming Weigh model can be in a real practical attack. In order to find that "point", we calculate the average power consumption in every trace point and plot the correlation coefficient between this average power consumption and the Hamming Weight of intermediate value corresponded to the point. As we can see in the Figure 4-13, 4-15 and 4-17, the trace points are extremely close to 1.0 at point 1309, 1589 and 2229, which means that the Hamming Weight of intermediate value at these points are very likely linear.

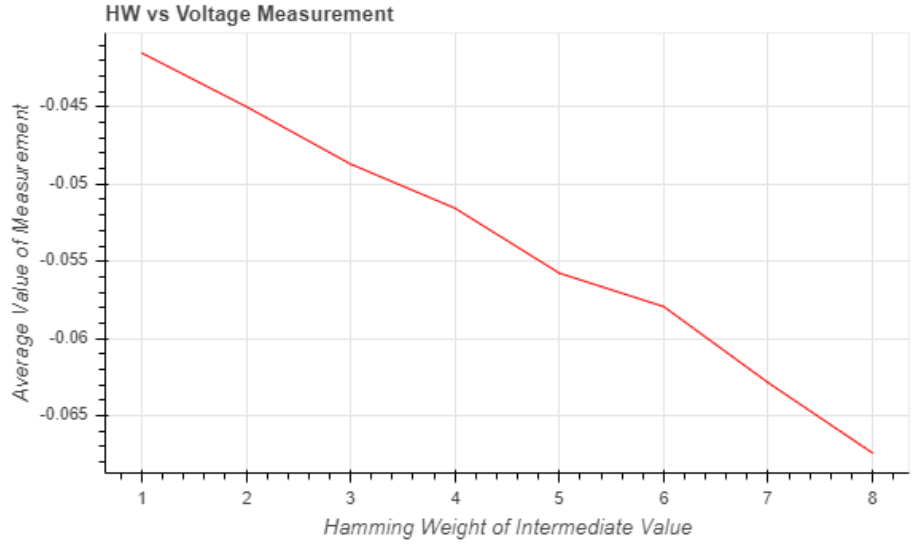


Figure 4-22. HW vs Power Consumption, STM32F303

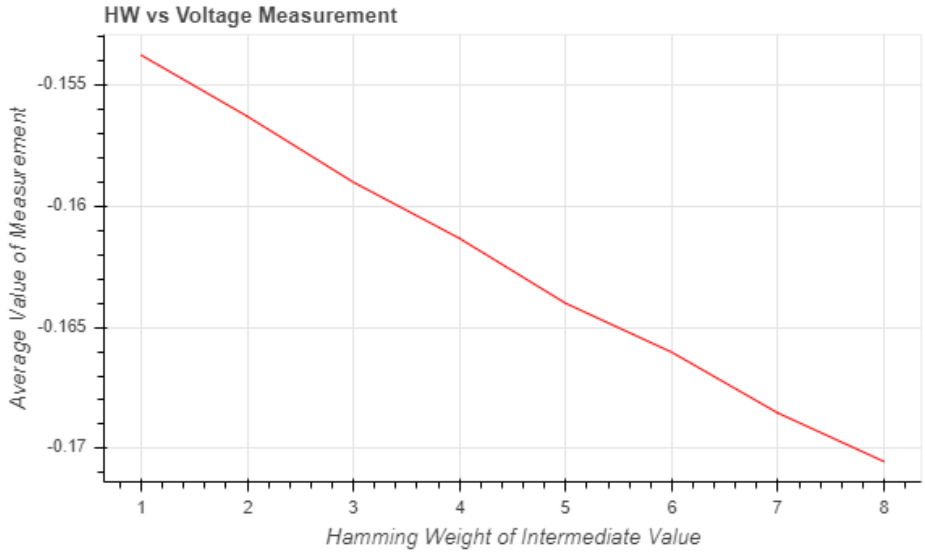


Figure 4-23. HW vs Power Consumption, STM32F071

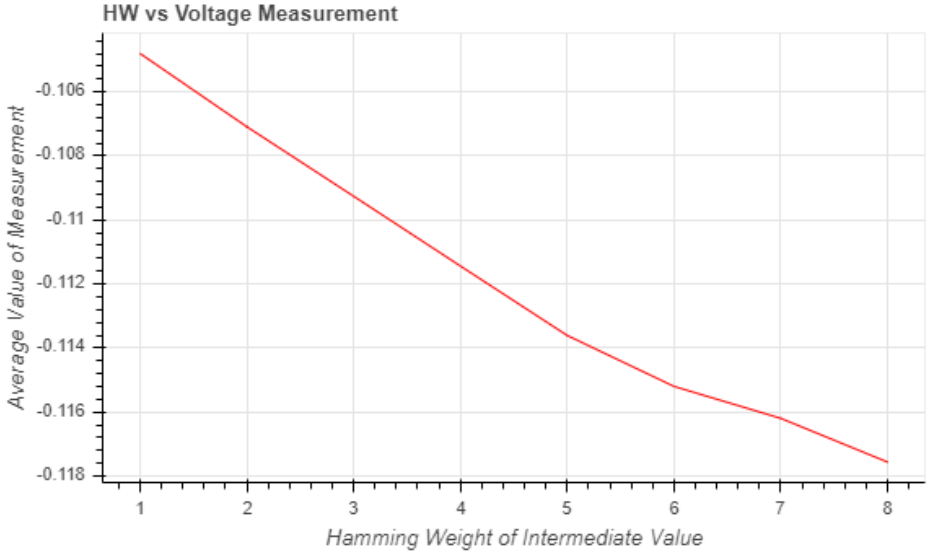


Figure 4-24. HW vs Power Consumption, XMEGA

For intuition and clarity, we plot the relationship between the average power consumption value of measurement and the Hamming Weight of intermediate value at these three selected sample points. As shown in Figure 4-22,4-23 and 4-24, we pick out

an interesting point which has the most correlation coefficient and calculate the average power consumption of the total recorded power traces. These average power consumption values correspond to the different Hamming Weight in a perfect linear relation. That means Hamming Weight information leakage at this point is higher, as we could extract some relative information of the secret key based on this. Also, if we take a closer look, we could find this “linear” line is not totally straight. Although the correlation coefficient is high enough, it is still not able to reach to 1.0. The information gets interrupted by the encryption process and the noise. However, we still could get a pretty good result from these “interesting” points which have the Hamming Weight information leakage.

Among 8000 power traces we measured and recorded, the quantity of the power traces of different Hamming Weight of intermediate value are not the same. Figure 4-25 depicts that HW 4 has the largest amount of power traces, in contrast, HW 0 and HW8 have the least amount of power traces. The distribution of power traces of Hamming Weight is more similar as the Gaussian Distribution. What happened in this situation is because the number of 0 and the number of 1 in HW 4 is equal. There is more chance that the hypothetical intermediate values have the same amount of 0 and 1.

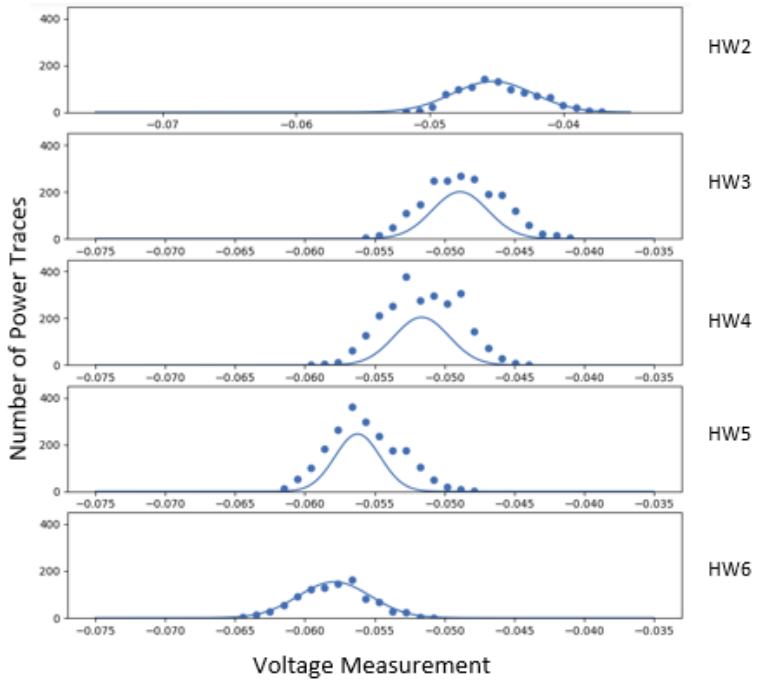


Figure 4-25.Measured and Matched Distribution model of STM32F303

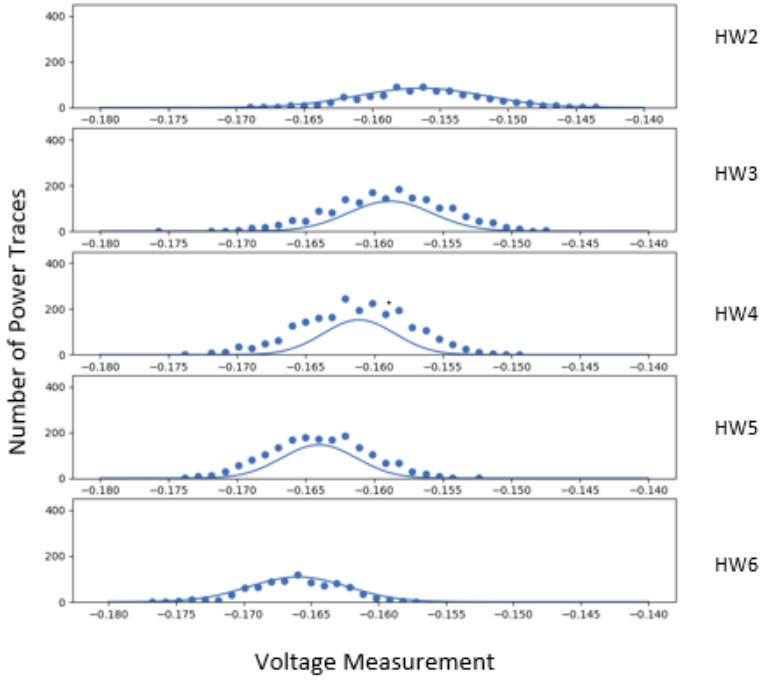


Figure 4-26. Measured and Matched Distribution model of STM32F071

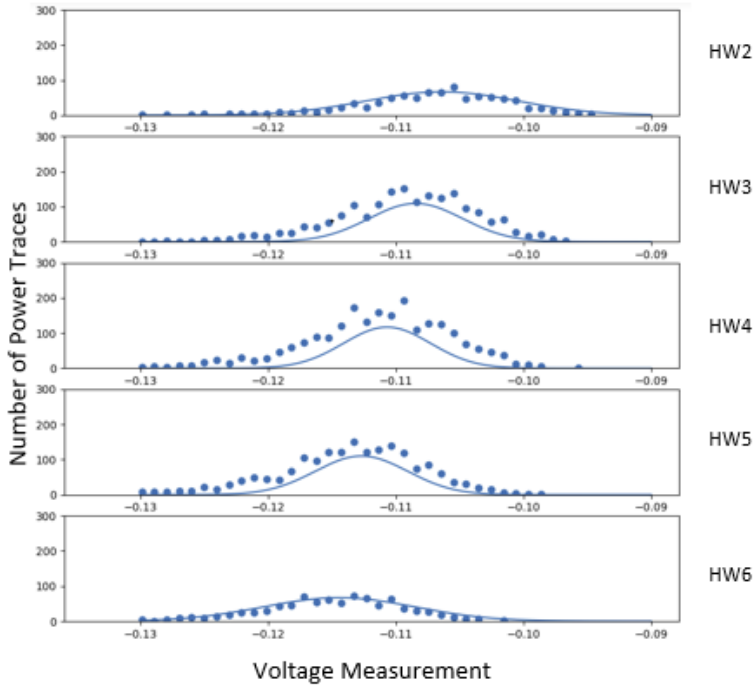


Figure 4-27. Measured and Matched Distribution model of XMEGA

We record 8000 power traces from the devices we are aiming to attack. For each Hamming Weight of intermediate value, we count how much occurrence the power traces are in different power consumption. The main goal for this analysis is to demonstrate how close the occurrence of the power traces in different power consumption matches with Gaussian Distribution.

Because of the lack of RAM in the Virtual Box we use to record power traces and analyze data, the number of the power traces we recorded in HW0, HW1, HW7 and HW8 is not enough to have a curve-fitting corresponding to them, as the reason we mentioned before. Here we focus on the rest of plots which are qualified to deliver the result. Therefore, as shown in Figure 4-25, 4-26 and 4-27, there are 5 plots demonstrated the relation between the occurrence of power traces and the measured power consumption, each of them corresponds to one of the Hamming Weight of intermediate

value. All the scatter dot lines exhibit the distribution of the occurrence of the recorded power traces in different power consumption. In every subplot, the curve-fitting using the Gaussian Distribution Model is drawn along with the corresponding scatter dot line.

As we explained before, the CPA attack is performing the Bayesian estimation of the "best key guess" and comparing the estimation with the measurement. In another word, we first bring up with all the possible secret key guess as likelihood function. By calculating the Pearson's Correlation Coefficient between the key guess and the measured power traces, the key guess corresponding to the largest correlation coefficient is the secret key we are looking for. Because of using Hamming Weight model to reveal the information leakage, closer the occurrence of power traces is to the Gaussian Distribution in each Hamming Weight Value, better the performance of Bayesian estimation will be.

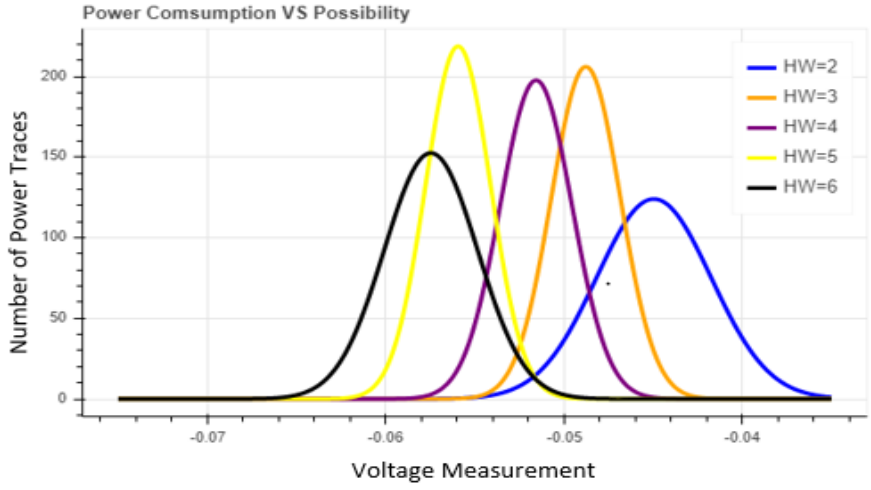


Figure 4-28. Matched Gaussian Distribution STM32F303

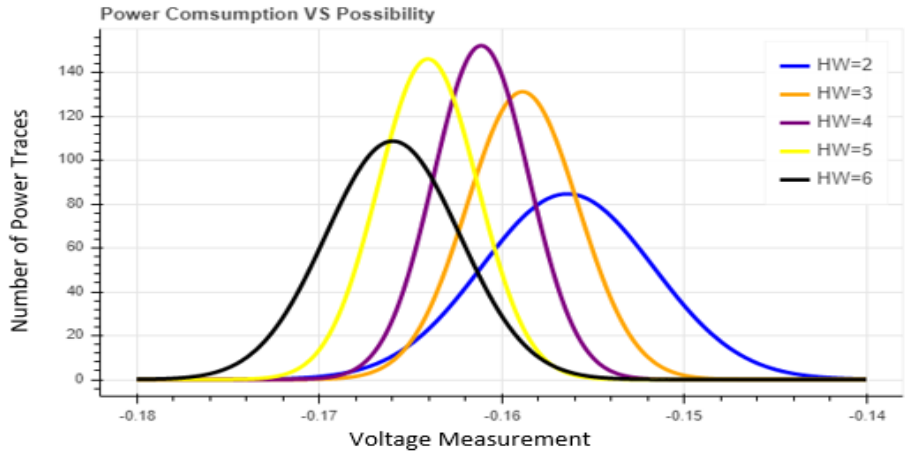


Figure 4-29. Matched Gaussian Distribution STM32F071

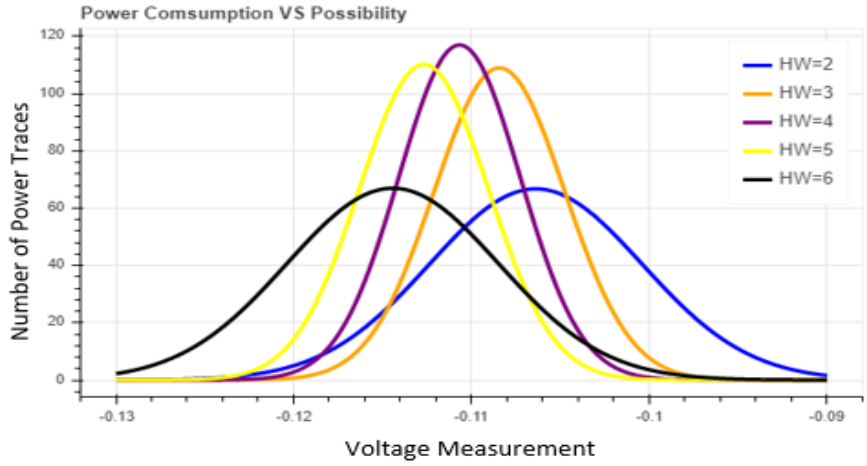


Figure 4-30. Matched Gaussian Distribution XMEGA

For clarity and intuitive, we extract the curve-fitting of Gaussian Distribution from Figure 4-25, 4-26 and 4-27. Apart from the lacking data in HW0, HW1, HW7, HW8, the patterns of HW2—HW6 match with Gaussian Distribution very well. The peaks of these curves follow the similar pattern with each other. Besides, the widest horizontal axis interval among these curves is still less than 0.03, which is acceptable as the attack condition.

Table 4-2. Statistic Data STM32F303

HW	MEAN (measured)	MEAN (matched)	STD (measured)	STD (matched)
2	0.0694	0.06935	0.002122	0.002380
3	0.0695	0.06940	0.002079	0.001473
4	0.0697	0.06964	0.002147	0.001449
5	0.0698	0.06969	0.002127	0.001570
6	0.0700	0.06993	0.002140	0.002404

Table 4-3. Statistic Data STM32F071

HW	MEAN (measured)	MEAN (matched)	STD (measured)	STD (matched)
2	-0.15629	-0.15638	0.004278	0.004715
3	-0.15901	-0.15884	0.004061	0.003042
4	-0.16132	-0.16110	0.003862	0.002621
5	-0.16400	-0.16402	0.003505	0.002731
6	-0.16602	-0.16595	0.003157	0.003672

Table 4-4. Statistic Data XMEGA

HW	MEAN (measured)	MEAN (matched)	STD (measured)	STD (matched)
2	-0.10710	-0.10639	0.005546	0.005979
3	-0.10925	-0.10843	0.005378	0.003661
4	-0.11144	-0.11068	0.005585	0.003411
5	-0.11361	-0.11268	0.005533	0.003621
6	-0.11520	-0.11443	0.005715	0.005959

As shown in Table 4-2, 4-3 and 4-4, we list all the mean values and standard variance values in both measured part and matched part. The difference between two part is slight, especially, in the match of mean values. However, in the match of standard variance

values, the matched values are slight smaller than the measured values at HW 3, 4 and 5. This difference illustrates that the number of power traces is a little bit less than we expect.

4.3 Performance of CPA based on HW Model with AWGN

As we illustrated before, the performance of CPA attack of HW model is in a good quality during the decryption without external noise. In this part of chapter, we add AWGN in the attacking system in order to evaluate the performance of CPA attack with noise. AWGN is often used as a channel model in which the only impairment to communication is a linear addition of wideband or white noise with a constant spectral density and a Gaussian Distribution of amplitude. The model does not account for fading, frequency selection, or dispersion. However, it produces simple and tractable mathematical models which are useful for gaining insight into the underlying behavior of a system before these other phenomena are considered. Before the plaintext enter the encryption device, there is an AWGN channel. Because of data corruption, the value of correlation coefficient is lower when the variance of AWGN goes up. However, we can reduce this corruption by increasing the number of traces we recorded. More number of recorded traces we calculate, the higher the value of correlation coefficient we get. As shown in Figure 4-31, we set up an AWGN channel before plaintext goes into the chips. The variances of the noise are from 0 to 2.0. The higher the added noise is, more corruption the data has. Before the variance of noise hit to 1.4, the number of traces remains in around 500. With the increment of the noise power, the number of traces we need to achieve a successful attack is increased exponentially. Because of the strong noise, the linear relationship between the intermediate value of HW model and recorded power consumption becomes weaker and weaker. Till last, because of the limitation of

the disk space, the data cannot be measured and analyzed.

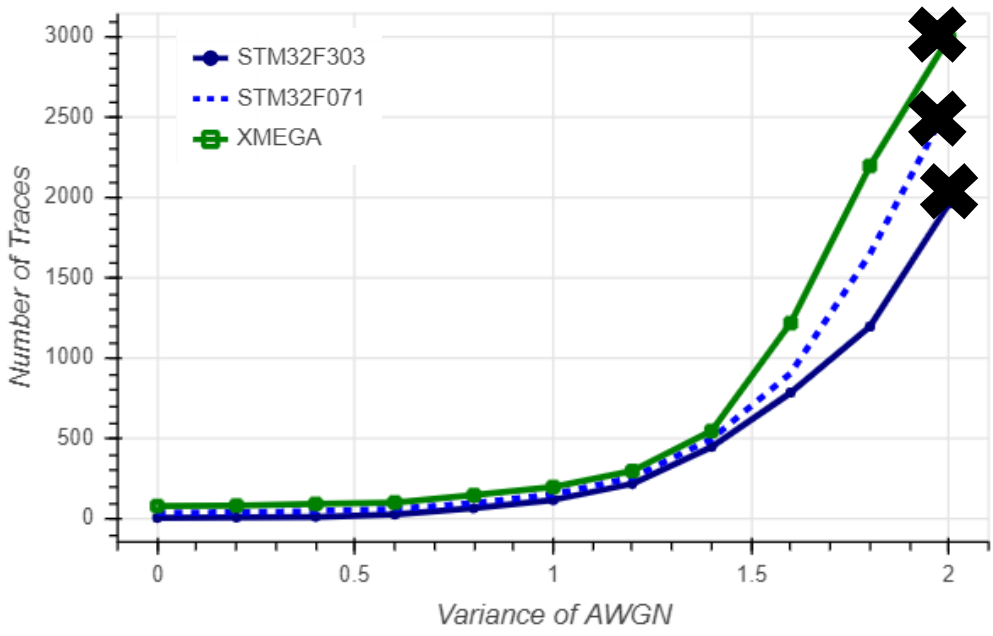


Figure 4-31.Variance of AWGN VS Number of Traces (“X” denotes that the number of traces cannot be calculated)

We take STM32F303 as an example to illustrate the relationship between voltage gain and attack success rate. We add an AWGN channel before the plaintexts enter the device. We set the variance of AWGN channel as 0.8 and the voltage gain of ADC from 5 to 75. As shown in Figure 4-32 below, in the noise free situation, the voltage changing has no influence on the attack. The number of traces we need to achieve a successful attack fluctuates around 10. However, in the case of AWGN, when the voltage gain is in a high level or a low level, the impact to the attack results goes up higher. When the ADC amplifies the signal, it amplifies the noise at the same time. When the voltage amplification exceeds a certain value or it is under a certain value, the noise become the dominant part, the attack result is compromised.

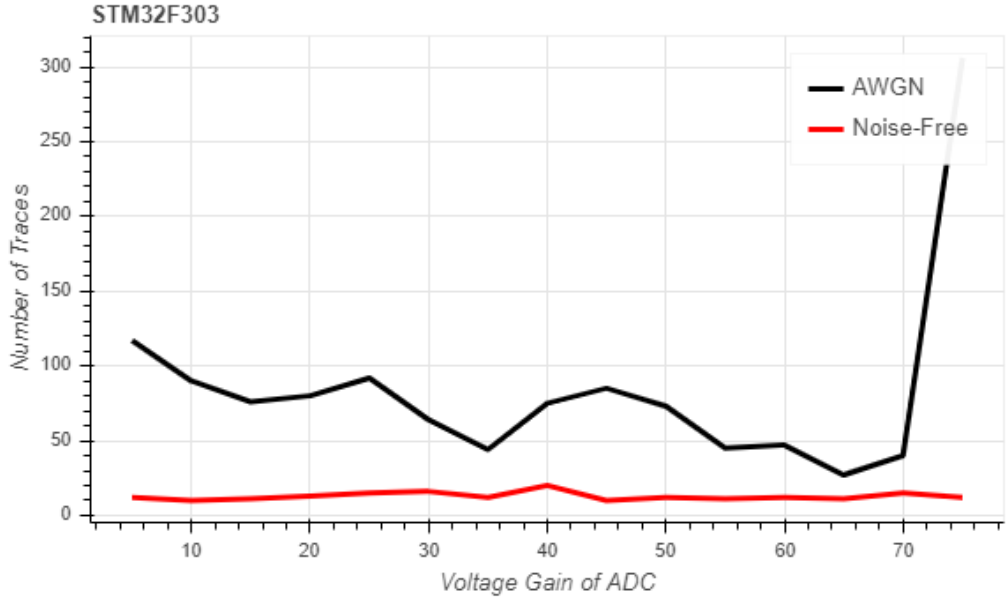


Figure 4-32.Voltage Gain of ADC VS Number of Trace

In a word, based on the figures above, we conclude that the noise resistance of CPA attack has a certain threshold. Even though we can reduce some of the noise impact by increase the number of traces we record, the attacking result is still compromised when the variance of noise exceeds the threshold. The voltage changing has influence on the attacking result as well. When the voltage level is either too high or too low, the noise will become the dominate part and the attacking result will collapse.

CHAPTER 5 CONCLUSION & FUTURE WORK

5.1 Conclusion

In this thesis, we introduce the encryption algorithm of AES and three methods (SPA, DPA, CPA) of side channel attack. We focus on the performance of CPA attack using HW model applied across three different devices (STM32F303, STM32F071 and XMEGA). As shown in chapter 4, the attack results against all these three devices is in a good quality without noise. Each of these three devices has at least one “interesting point” that has information leakage. The relationship between the intermediate value of HW model and measured power consumption is in a linear way at that point. The distribution of the recorded power consumption matches Gaussian Distribution very well in each different value of HW. That means HW model has a good description of the power consumption during the attack and the CPA attack is a success based on this model description.

In addition, we conclude that the noise resistance of CPA attack has a certain threshold. Even though we can reduce some of the noise impact by increase the number of traces we record, the attacking result is still compromised when the variance of noise exceeds the threshold. The voltage change has influence on the attacking result as well. When the voltage level is either too high or too low, the noise will become the dominate part and the attacking result will collapse.

5.2 Future Work

In this thesis, we first give a demonstration of the performance of CPA attack using HW model across multiple devices and calculate the best distribution model of the measured

power consumption. We conclude a general attacking principle and distribution model for CPA attack. However, because of lack of the memory space in the Virtual Box we use to perform the experiment, the largest data collection we can save is 9000 traces. This number of traces is not enough to perform calculation of distribution in HW 0, 1, 7 and 8. We have to discard these 4 values of HW when we calculate the distribution of power consumption. Besides, when we add AWGN into the system, more and more traces are necessary to achieve a successful attack. However, after exceeding a certain noise threshold, the attack result collapses because of too many recorded traces. In the future work, we will focus on collecting more traces and corresponding data to have a more thorough concept of CPA attack with and without noise included.

Another issue that deserve to be investigated is the performance of CPA attack using Hamming Distance (HD) Model. As another popular hypothetical power model, HD model is as important as HW model in the attack. In the future work, we will take HD model into consideration and calculate the performance of it.

REFERENCES

[1] Kocher, P. C. (1996). Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1109, 104-113.

[2] Lyu, Y., & Mishra, P. (2018). A Survey of Side-Channel Attacks on Caches and Countermeasures. Journal of Hardware and Systems Security, 2(1), 33-50.

[3] Van C., J., De S., B., & De Bosschere, K. (2017). Adaptive Compiler Strategies for Mitigating Timing Side Channel Attacks. IEEE Transactions on Dependable and Secure Computing, PP (99), 1.

[4] Molter, H., Stöttinger, G., Shoufan, M., &Strenzke, A. (2011). A simple power analysis attack on a McEliece cryptoprocessor. Journal of Cryptographic Engineering, 1(1), 29-36.

[5] Changhai O., Zhu W., Degang S., Xinping Z., & Juan A. (2016). A New Efficient Interesting Points Enhanced Electromagnetic Attack on AT89S52. 2016 IEEE International Symposium on Electromagnetic Compatibility (EMC), 2016, 176-181.

[6] Genkin, D., Shamir, A., &Tromer, E. (2017). Acoustic Cryptanalysis. Journal of Cryptology, 30(2), 392-443.

[7] Yu K. (2009). Security strategy of powered-off SRAM for resisting physical attack to data remanence. Journal of Semiconductors, 30(9), 5.

[8] Cai, Z., Wang, Y., & Li, R. (2013). Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8300, 441-449.

[9] FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001.

[10] Brier, E., Clavier, C., & Olivier, F. (2004). Correlation power analysis with a leakage model. Cryptographic Hardware and Embedded Systems - Ches 2004, Proceedings, 3156, 16-29.

[11] Li, H., Yu, F., & Wu, K. (2011). Enhanced correlation power analysis attack against trusted systems. Security and Communication Networks, 4(1), 3-10.

[12] Kocher, P., Jaffe, J., Jun, B., & Rohatgi, P. (2011). Introduction to differential power analysis. Journal of Cryptographic Engineering, 1(1), 5-27.

[13] Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks revealing the secrets of smart cards. New York: Springer.

[14] Zadeh, A., &Heys, H. (2015). Application of Simple Power Analysis to Stream Ciphers Constructed Using Feedback Shift Registers. The Computer Journal, 58(4), 961-972.

[15] Chadha, R., & Bhasker, J. (2013). An ASIC low power primer: Analysis, techniques and specification. New York, NY: Springer.

[16] Sklyarov, V. (2014). Synthesis and optimization of FPGA-based systems (Lecture notes in electrical engineering; v. 294). Cham: Springer.

[17] Barr, K. (2000). ASIC design in the silicon sandbox a complete guide to building mixed signal integrated circuits (McGraw-Hill's Access Engineering). New York: McGraw-Hill.

[18] Allen, P., &Holberg, D. R. (2012). CMOS analog circuit design (Third ed., Oxford series in electrical and computer engineering). New York; Oxford: Oxford University Press, USA.

[19] Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES--the Advanced Encryption Standard. Berlin; New York: Springer.

[20] "Advanced Encryption Standard, NIST FIPS PUB", "National Institute of Standard and Technology", vol. 197, 2001.

[21] Carlet, C., Faugère, J., Goyet, C., & Renault, G. (2012). Analysis of the algebraic side channel attack. Journal of Cryptographic Engineering, 2(1), 45-62.

[22] Roy, D., Datta, P., & Mukhopadhyay, S. (2015). Algebraic cryptanalysis of stream ciphers using decomposition of Boolean function. Journal of Applied Mathematics & Computing, 49(1-2), 397-417.

[23] Moon, J., & Park. (2018). IoT application protection against power analysis attack. Computers and Electrical Engineering, 67, 566-578.

[24] Chen., H., & Chen, H. (2011). A novel algorithm of fingerprint encryption using minutiae-based transformation. Pattern Recognition Letters, 32(2), 305-309.

[25] Hermassi, H., Rhouma, R., &Belghith, S. (2013). Improvement of an image encryption algorithm based on hyper-chaos. Telecommunication Systems, 52(2), 539-549.

[26] Steef, A., Shamma, M., &Alkhatib, A. (2016). RSA algorithm with a new approach encryption and decryption message text by ascii.

[27] Liu, C., Zhou, Y., Xiao, Y., & Sun, G. (2011). Encryption algorithm of RSH (Round Sheep Hash). Information Technology Journal, 10(3), 686-690.

[28] Kocher, P., Jaffe, J., & Jun, B. (1999). Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1666, 388-397.

[29] Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks revealing the secrets of smart cards. New York: Springer. Page 102-103

[30] Chari, S., Rao, J., & Rohatgi, P. (2002). Template attacks. Cryptographic Hardware and Embedded Systems - Ches 2002, 2523, 13-28.

[31] Wang, A., Wang, Z., Zheng, X., Wang, X., Chen, M., Zhang, G., & Wu, L. (2015). Efficient collision attacks on smart card implementations of masked AES. Science China Information Sciences, 58(5), 1-15.

[32] Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks revealing the secrets of smart cards. New York: Springer. Page 105-111

[33] Souvignet, &Frinken. (2013). Differential Power Analysis as a digital forensic tool. Forensic Science International, 230(1-3), 127-136.

[34] Chen, C., Eisenbarth, T., Von Maurich, I., &Steinwandt, R. (2015). Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9092, 538-556.

[35] Mi š kovský, Kub á tov á , &Novotný. (2017). Influence of passive hardware redundancy on differential power analysis resistance of AES cipher implemented in FPGA. Microprocessors and Microsystems, 51, 220-226.

[36] Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks revealing the secrets of smart cards. New York: Springer. Page 121-123

[37] Hamming, R. (1980). Coding and information theory. Englewood Cliffs, N.J.: Prentice-Hall.

[38] Liu, Goto, &Tsunoo. (2011). Correlation Power Analysis with Companding Methods. Procedia Engineering, 15, 2108-2112.

[39] Li, H., Wu, K., & Yu, F. (2011). Enhanced correlation power analysis attack against trusted systems. Security and Communication Networks, 4(1), 3-10.

[40] Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks revealing the secrets of smart cards. New York: Springer. Page 112-113

[41] STM32F071. (2015, September 27). Retrieved June 24, 2019, from: st.com

[42] STN32F303. (2015, September 27). Retrieved June 24, 2019, from: st.com

[43] O' Flynn, C., & Chen, Z. (2014). Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8622, 243-260.

[44] ATSAM3U2C, Microchip. (2012, December 31). Retrieved June 24, 2019, from: microchip.com

[45] SPARTAN - 6. (2019, April 16). Retrieved June 24, 2019, from: xilinx.com

[46] Huang, H. (2014). The Atmel AVR microcontroller: Mega and XMega in Assembly and C. Clifton Park, NY: Delmar.