November 2016

Test Vector Leakage Assessment Development

Romuald S. Valme
Worcester Polytechnic Institute

Test Vector Leakage Assessment Development

Major Qualifying Project

Advisors:

Professor T. Eisenbarth
Joseph Chapman

Written By:

Romuald Valme

A developmental endeavor with conjoined efforts of
The MITRE Corporation and
Worcester Polytechnic Institute

A student-based undertaking submitted in the form of an Major Qualifying Project at the Worcester Polytechnic Institute.

A-2016

Abstract

Devices with a lack of countermeasures, which are defenses put into place to thwart an assailant, can be subject to physical attacks. These attacks can lead to the extraction of sensitive information such as keys that are used in cryptographic operations to secure for example intellectual property. Test Vector Leakage Assessment (TVLA) aims at being able to provide detection of information leakage using statistical analysis. This work presents multiple implementations of TVLA that aid in the testing of these systems and countermeasures, which is crucial in keeping information secure.

Contents

List of Figures

Figure **Page**

Chapter 1

Introduction

In 1999, Paul Kocher, detailed his ability to directly determine power consumption measurements collected during cryptographic operations [2]. Using this information he could obtain information about device operations and potential encryption and decryption key material. Kocher went on to use various forms of power analysis to determine entire keys and important intermediates. His work spawned new forms of hardware attack that has now stretched over a decade of new research and countermeasures. Researchers at Jiao Tong University were able to crack the encryption on 3G and 4G sim cards. Using side-channel analysis, researchers tracked the power levels of the cards on a oscilloscope [3]. Through analyzing the protocol and correlating the results through a SIM card reader, their team was able to not only crack the encryption, but clone SIM cards. This capability allowed for impersonations of accounts such as Alipay, where account funds could be withdrawn and passwords could be changed. This attack is due to the fact that developers of the cards were reliant only on software security for these devices and implemented no hardware countermeasures.

In society today, data and information are fundamental to many factors of life such as health care, finances, and education. However, devices in these areas of life are vulnerable to devastating hardware attacks. As described, educated hackers are more than capable of manipulating these systems to their will. Countermeasures and protection for these devices must be established and implemented. Thankfully, there are several forms of countermeasures such as masking data or obfuscating power consumption signals for confusion. Verifying and making sure these countermeasures are working as intended is essential. One must be able to understand these attacks in as many ways as possible. Much work must be done in testing these devices.

One approach to side-channel testing is to perform a side-channel attack on the device known as Differential Power Analysis (DPA). However, performing such an attack has it's drawbacks. One previous work on testing systems and countermeasures for their security shows how statistical analysis of power traces can determine if ones system is vulnerable to hardware attacks through side-channel analysis [4].

The goal of this Major Qualifying Project (MQP) is to develop a statistical analysis tool that allows an individual to see if ones system is subject to hardware attacks such as DPA. In this MQP this was accomplished by first learning more about the statistics behind such a tool. This learning was done through a literature review. Subsequently the analysis framework was studied since it is the space where the tool will live. Following, development began on the tool. Numerous variations of the tool were made proving its flexibility and use.

The tool is not only capable of detecting if a system is subject to some form of information leakage, but is also capable of detecting higher moment information leakage. The tool was tested at various moments with precise data sets to observe behavior, and behaved as expected. The tool is also able to function despite moderate to severe environmental noise.

While developing this tool, much understanding was gained on improvements that could be made to the system. A few recommendations include improved memory input and output handling, cache detection of alterations made to code, and improved fragmentation of stored data.

Chapter 2

Background

2.1 Overview

This background section provides critical information in respect to security, cryptography, vulnerabilities of current systems, and ways to detect those vulnerabilities. One must understand these concepts and ideas in order to have a proper understanding of the design, work, and results achieved in this project. This section provides a clear path to comprehending the methodology.

2.2 Protecting Information Exchange

When exchanging information, often times such information must be kept secret between desired parties. This circumstance is especially important in situations such as when critical intellectual property must be protected, or when there is a sensitive message that must only be read by a particular individual. In order to protect this information many cryptographers have devised certain methods that can provide security.

2.3 Cryptography

Cryptography has four information security objectives from which all others are derived. They are the following: confidentiality, which is a service that only allows those who are authorized to have certain information, data integrity, which disallows the unauthorized tampering or modification of data, authentication, which allows one to identify individuals as themselves, and non-repudiation, which prevents denial of service [5].

2.3.1 History

First Occurrences

One of the earliest instances of cryptography traces back to 1900 BC where in Egypt an inscription had been carved into the tomb of a nobleman [6]. Some of the symbols the scribe used were unusual [7]. Apparently the individual had altered certain symbols in the message. However, his reasoning was in order to make the message look more distinguished.

Caesar Cipher

Around 100 BC, during a time of war Julius Ceaser devised a cipher where he would shift a message's letters by a certain number, making the message intelligible [1]. This cipher was one of the tools that aided him in his Civil War against Pompey enabling him to gain power in Rome. Caesar was known by shifting his letters by 3. Figure 2.1 explains this in more detail.

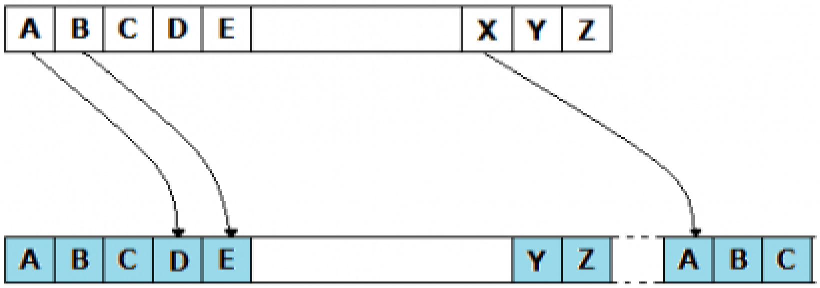

Figure 2.1: Caesar Cipher Shifting Mechanism (A Brief History of Cryptography, 2016)

Substitution Cipher

During the 16th century Mary Queen of Scots had the desire to assassinate Queen Elizabeth I. In order to communicate her intent with her partners, she encrypted these messages using a substitution cipher. Each letter would be substituted with a corresponding letter in a "cipher alphabet". Figure 2.2 shows the nomenclature that Mary Queen of Scots used. However, by this time in history letter frequency analysis had already been a well-known form of attack.This resulted in Elizabeth I capturing and cracking these messages. Mary Queen of Scots was tried, found guilty and and later executed [8].

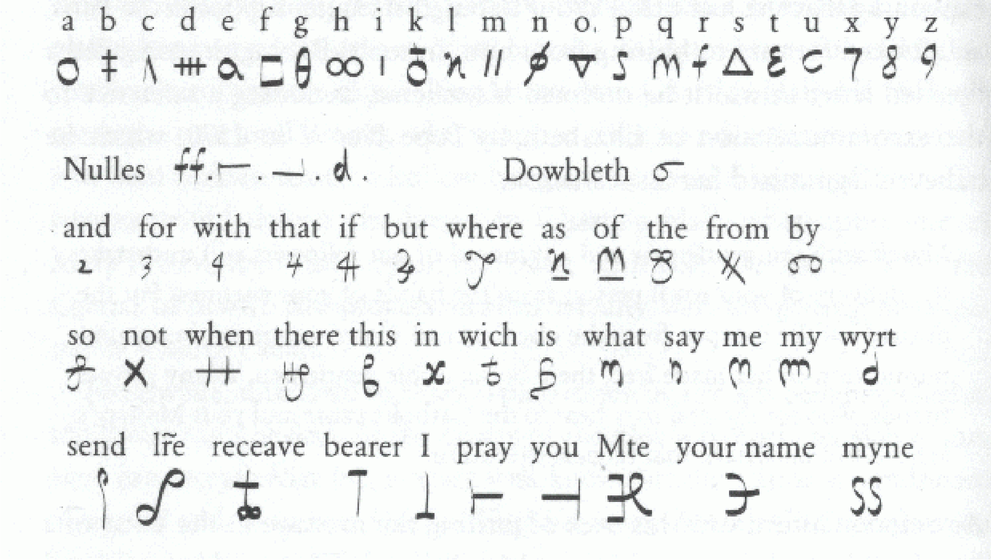
a b c d e f g h i k l m n o p q r s t u x y z
Nulles
Dowbleth
and for with that if but where as of the from by
so not when there this in wich is what say me my wyrt
send lr̃e receave bearer I pray you Mte your name myne

Figure 2.2: Mary Queen of Scots Substitution Cipher Nomenclature

Vigenere Cipher

Another important cipher that was devised during the 16th century is the Vigenere Cipher. This cipher is similar to the Caesar cipher but differs because it shifts each letter independent of the other. Each letter has a shift of its own rather than all letters of the message having the same shift. More emphasis is placed on the actual key than the system. The key repeats multiple times to the length of the message and provides more security than the Ceasar Cipher. Figure 2.3 uses the key "CRYPTO" which is repeated over the length of the message m, thus creating the cipher text c [1].

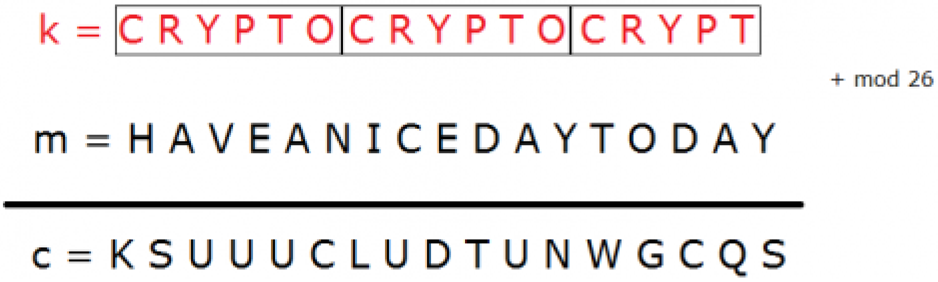

Figure 2.3: Vigenere Cipher Example [1]

2.3.2 Modern Applications of Cryptography

In this digital age most devices that individuals interact with on a daily basis use some form of cryptography. Whether it be the chip and pin system used by your bank and credit card, or the certificates used to verify the identity of a website. Figure 2.4 shows several different technologies that in some way all use cryptography. This fact illustrates just how much this form of security inundates the typical individual's life.

Figure 2.4: Devices that use Cryptography

2.3.3 Weaknesses of Early Ciphers

As hinted above early ciphers had many weaknesses. The ciphers above are all vulnerable to letter frequency analysis. A technique where one uses the most common letters used in the alphabet to analyze which letters may correspond to other letters.

The Ceaser Cipher is highly susceptible to a brute force attack one most only try 25 different variations of the message in order to correctly guess it. The Vigenere Cipher only improves on that to the length of your message. Granting 26^n possible different combinations for your message, where n is the number of characters in your message. This difference does provide some improvements however with computers today often times these messages can be deciphered easily.

2.3.4 Modern Cryptography Systems

In order to protect the devices and information that are used today, stronger forms of security and cryptography were needed. This led individuals to developing Symmetric and Public Key Cryptography systems in order to secure information. The following sections will describe these innovations in cryptography.

Symmetric Algorithms

Symmetric Algorithms are named as such because the same cryptographic key is used during the encryption and decryption process. This feature can be seen as an advantage or disadvantage. Since multiple parties will have the same key the authentication of who sent which message cannot be done. However, often times as due to the implementation of symmetric algorithms they are much quicker then asymmetric algorithms. These algorithms are typically used when there is a central controlling authority such as a government or military. They can live fine with symmetric protocols if there is an efficient way of agreeeing on keys. The Kerberos protocol is an example. It can run only with symmetric cryptography and is widely used for protecting corporate networks [9].

These types of ciphers can be implemented various ways. The two main ways are as stream ciphers and block ciphers. In block ciphers various bits are encrypted together and then usually padded to some length. In a stream cipher usually each byte is encrypted on its own. There are various symmetric key algorithms [10].

The most widely used symmetric algorithm known as the Advanced Encryption Standard (AES) was created through the 1997-2000 competition and was organized by the National Institute of Science and Technology. A contender for the AES competition The algorithm that won Rijndael became known as AES [11]. A popular algorithm used before AES is the Digital Encryption Standard (DES). It was later determined to be no longer secure, and now individuals have modified it and use it in the algorithm Triple DES (3DES). The Rivest Cipher4 is a well known cipher known for its use in Transport Layer Security [12].

Public Key Systems

In public key systems also known as asymmetric cryptography, a pair of keys are used. The two different keys generated are known as the public and the private key. One is widely available and one must be kept only for an intended individual. In these systems authentication is available, as only a single individual will typically have the private key proving they sent a message [13].

In these systems the public and private key can both be used for encryption and decryption. However, it is important to note that only the private key can decrypt the public keys encrypted messages, and only the public key can decrypt the private keys messages.

One popular and intended use of public key systems is authentication this process typically consists of hashing a message into a digest. Afterwards encrypting this output using the private key. In this scheme the message is typically known and the only need of the protocol at this step is authentication [5]. Exclusively to public key systems are digital signatures, which is another method of providing authenticity for a document or message and key exchanges, which allow for secure transfer of keys used for encryption and decryption.

Popular algorithms are RSA, which stands for the names of its creators Rivest, Shamir, and Adleman. El Gamal and the Diffie-Hellman are commonly used for key exchanges. The Digital Signature Standard as its name implies is commonly used for digital signatures.

2.4 Security Hardware Vulnerabilities

2.4.1 Overview

Cryptographic systems are vulnerable to physical attacks. Attacks through techniques such as DPA make use of the natural characteristics of a device, and are possibly vulnerable despite the security that exist in software. These attacks can be devastating causing weird and unintended behavior in devices or even to the extent of leaking whole keys, resulting in successful extraction of secret messages.

2.4.2 Side-Channel Analysis

Side-Channel Analysis can occur in most any systems. Side-channel information is simply unintended information released by a system due to its physical characteristics. An example is of a traditional safe or lock, which releases unintentionally information on whether a number tried is correct through sound.

Digital systems have certain characteristics which result in side-channel analysis being feasible.This information is released in digital systems in various ways. It can be released through the power consumed, the heat released, acoustic vibrations, and even the electromagnetic waves it creates. Figure 2.5 shows an example setup for observing power consumption on a Sasebo-GII FPGA, while performing a basic AES-128.

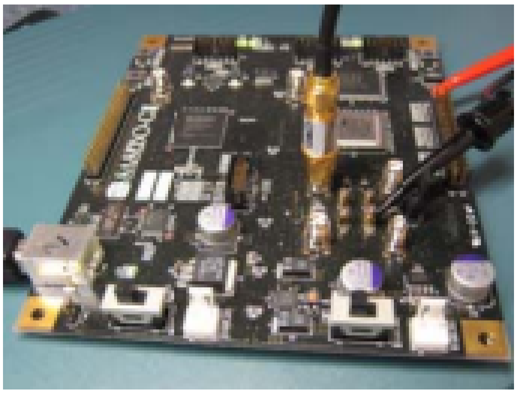

Figure 2.5: Example Setup on Sasebo-GII FPGA

Resultingly there are a plethora of different attacks that can be done. On a system a few notable attacks discovered throughout history are Timing attacks, Power Consumption Attacks, which stem out into Simple Power Analysis (SPA), and DPA, or Differential Fault Analysis (DFA) Attacks, also known as Fault Injections. These various attacks can even be combined to create devastating results [14].

Simple Power Analysis

SPA is a basic analysis of the traces produced through observation. In an MQP in 2013 titled *Electromagnetic Side-channel Analysis on an Intel Atom Processor*, students were able to perform SPA and even read out the used key from the technique [15]. This technique can be performed on the trace in Figure 2.6. If one knew nothing about the algorithm being used, using SPA one could see that the trace has 10 repetitive structures in it.

This information hints at the algorithm being used to be AES-128 since it has 10 rounds where the same operations are being done on inputted data. With this information more advanced techniques can be used to extract even more sensitive information.

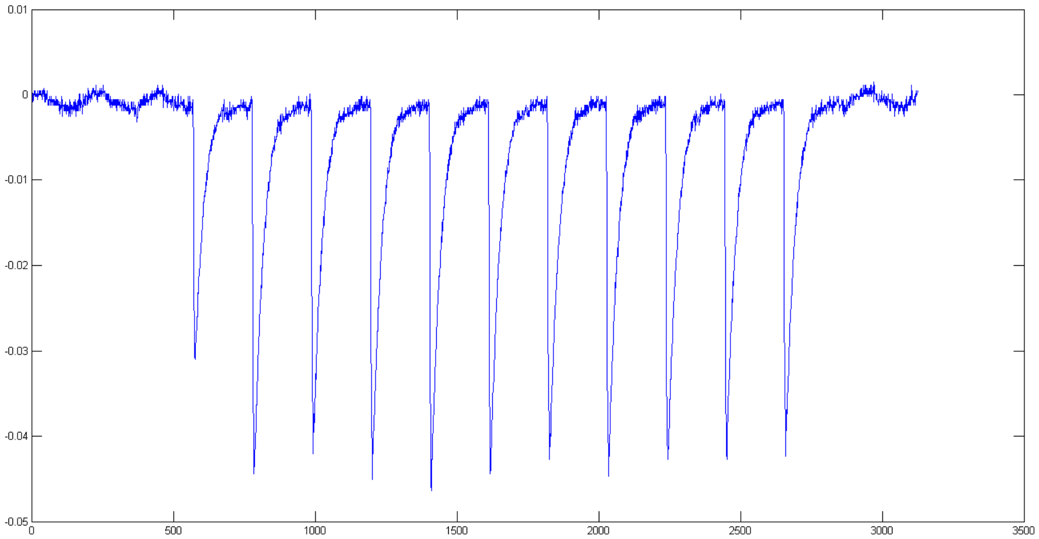

Figure 2.6: AES Trace on Sasebo-GII FPGA

Differential Power Analysis

DPA is a much more powerful attack. This attack can lead to obtaining key information about a system. DPA is a statistical method wherein sets of measurements to identify data-dependent correlation occur. In a DPA a set of traces is divided into different subsets. The difference of the averages of these subsets is computed.

The subsets are divided based off of a selection function.The selection function is typically determined based off of an educated guess as to a possible intermediate state in the system. If the outputted vector shows a significant spike a cryptanalyst knows the selection function output is correlated to a value operated on by the device under test (DUT).

Fault Injection

This type of attack is involved with changing the different characteristics of a system. Such as changing the power supplied to a device, or messing with the clock of a device [14]. These can cause lines of code to be entirely skipped. Fault injection is a manipulation of the internal state of a device by adding some type of outside factor or change.

2.5 Detecting Vulnerabilities

2.5.1 Directed Tests and Attacks

One method of choice used in knowing if a system is secure is performing some sort of side-channel attack such as DPA on the device. However, this type of vulnerability detection can prove to be very inefficient as the correct hypothesis must be made for the selection function. This can lead to false positives, where a system is thought to be secure but in actuality is vulnerable.

This unknown factor or educated guess that must be made when testing a system proves to be very difficult to deal with. The complexity of implementing such designs can prove to be difficult as well. As assumptions must be made about intermediate states of the device. Repetition of these attacks on a system is resultingly common, because the correct guess must be made. In order for there to be a "hit". This repetition is time consuming, which can be an issue in time sensitive situations.

This project deals with implementing a system that is able to bypass these issues creating a tool that tells a user if there system is leaking critical information and is vulnerable to side-channel attack.

2.5.2 Test Vector Leakage Assessment

A statistical tool, which can be used to detect information leakage in devices is TVLA. This approach tries to detect if sensitive intermediates have significantly impacted the measurement data. The null hypothesis of TVLA is that the mean and variance of the two sets of power traces are the same meaning that the sensitive intermediates had no impact on them [4]. In this test the Welch's T-test is used to determine whether a data set provides enough information to prove the null hypothesis false. Equation 2.1 below shows Welch's T-test.

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{2.1}$$

There are two main categories of detection through TVLA. There is general and specific detection. The general case is typically composed of a fixed and random data set as the inputs to the cryptographic operations. The term fixed is meant to imply that over the course of the collection of the traces, these fixed traces were derived using an unchanging data set. This unchanging data set underwent the encryption process where the traces were collected. The term random implies a similar meaning. The difference is that these random traces were derived from a randomly changing data set.

The goal of this general fixed vs. random test is to detect if there is any leakage depending on the input data or key. Keeping one data set fixed and randomly generating the other allows one to achieve this goal. A general test showing leakage however, does not necessarily guarantee a key attack. It only indicates the possibility of an attack [4].

A specific test usually is targeting an intermediate in a cryptographic algorithm. This intermediate could be an attack point for determining a key. Typical intermediates are s-box outputs, round outputs, or the XOR during a round input or output. This test can be achieved using a random vs. random data set. These data sets ensure that there is no leakage depending on the input data.

2.5.3 Welch's T-test

The goal of this t-test is to determine whether two samples have the same mean. This t-test is based off of the Student's T-test, created by William Gosset [16]. This test is used to detect how different two sets are from each other. In figure 2.7 describing Welch's T-test $\overline{X}$ refers to the mean of a distribution. In the numerator the mean of the two distributions are being differenced. In the denominator the s^2 refers to the variance of a distribution as variance is the standard deviation squared. The variance of a distribution is being divided by the total number of samples in that set N, and then being added to the equivalent value for the other set.

2.5.4 Matched Pairs Test Vector Leakage Assessment

A variation on TVLA, allows for results to be correctly portrayed despite a good amount of environmental change or fluctuations. In the classical TVLA these environmental factors strongly effect the mean and variance of the two sets of power traces, and attribute greatly to them making it seem as though a system is secure, when it is actually vulnerable. Matched Pairs TVLA allows one to detect the differences in mean and variance despite these factors.

In this implementation each trace in a group is given a pair in the other group. Once collected a difference on them is done with their pair, the resultingly set of traces is known as the difference. A paired form of the Welch's t-test is then performed on the mean and variance of the set of traces. Equation 2.2 shows this paired Welch's t-test [17].

$$t_p = \frac{\overline{D}}{\sqrt{\frac{s^2}{N}}} \tag{2.2}$$

2.5.5 Countermeasures

There are three popular types of countermeasures that can be implemented in a system to defend against side-channel attack. These countermeasures perform best in combination with each other. They are known as hiding, masking, and leakage resilience. The following paragraphs are dedicated to explaining and providing insight into these countermeasures.

Hiding

This type of countermeasure tries to make it difficult to measure leakage by generating points of known leakage with some form of noise. Other than noise one can try using shielding which consists of putting metal shielding around components to prevent detection of certain electromagnetic emissions. One can also try to use dummy insertions, which entails adding in blocks of code that do not have to do with sensitive information [14]

The downfall to hiding however, is that often times if physical access to the device is granted shielding can be removed, as well for noise and dummy operations adding more traces to the attack can overcome them.

Leakage Resiliency

A common form of leakage resiliency is key cycling. This process entails frequently changing the key to deterministic values. The same key value is only used for a certain amount of time to reduce the number of traces obtained.

An implementations can include changing the cryptographic protocol for a system, not being able to be performed backwards, and the performance of the device being severely impacted. These facets of an implementation only add to its difficulty.

Masking

In masking the goal is to break the statistical significance of the algorithm from the values processed on in the algorithm such as the key or plain text. One can split the sensitive information into two shares for example. Operations are then performed on the shares individually. These implementations however are subject to higher order attacks [18].

2.5.6 Statistical Moments

In statistics, a data set has certain characteristics unique to it. These characteristics can be seen in the shape of the Probability Density Function (PDF) of the data set. The mean of the data set is known as the first moment. The mean is calculated by summing the data set, and dividing it by the number of values in the set. The variance of the data set is known as the second central moment. The variance is calculated by taking the difference from the mean for each sample in the set and squaring that value. The squared value for each sample is then summed for all values in the set and divided by the total number of values in the set. Equation 2.3 portrays the variance calculation.

$$variance = \frac{\sum(x_i - \bar{x})^2}{n} \tag{2.3}$$

For higher order moments the central or standardized moment are typically used as they do not affect the location of the distribution in the PDF. They only affect its shape and spread [19].

The third standardized moment is the skewness and it is seen through the lopsidedness of the PDF, whether it is more dense on one side then the other. The fourth standardized moment is the kurtosis. The kurtosis is seen by the shape of the tails at each side of the PDF.

2.5.7 Higher Moment Test Vector Leakage Assessment

Higher moment TVLA can be used to detect leakages despite masking using multiple shares. Higher moment TVLA is achieved by taking into account the mean of the groups of traces. One also must raise each trace to the power of whatever order is desired. Using the center product for the calculation of the higher orders can reduce computation complexity thus improving efficiency and speed. Equation 2.4 is an equation explaining how this might be implemented [17].

$$D = \left[(L_A - \bar{L}_A)^d - (L_B - \bar{L}_B)^d\right] \tag{2.4}$$

2.5.8 T-private

A t-private design provides circuit level information protection. For a masked design this protection is achieved by separating the input into $t + 1$ parts called shares [20]. T-private circuits are a masking countermeasure, and the more shares the higher security [21].

2.5.9 Summary

Concluding, the growth of cryptography has been ongoing for thousands of years. In the onset of technological advancement digital information security has become a necessity in numerous facets of life. In theory and practice however, systems have been shown to be devastatingly vulnerable to physical attacks. These attacks can be combated through countermeasures, but these countermeasures should be verified and tested with tools such as TVLA.

Chapter 3

Methodology

3.1 Overview

In order to detect potential information leakage in electronic devices, In this work a tool was created to implement various versions of a TVLA methodology. This tool provides a statistical vector that yields critical information about the security of one's system.

In order to create this tool, existing TVLA designs and implementations were examined. From these designs, multiple variations of TVLA that would work within the MITRE side-channel analysis framework were designed. The first phase included designing and implementing a basic fixed vs. random TVLA system and a more advanced matched pairs implementation of TVLA. A higher moment testing procedure was designed.

This TVLA system will be used by the MITRE Corporation in future endeavors, so this system needs to function without mishaps. Phase 2 included testing and verifying that this system worked without mistake. Various tests were run on the system for efficiency and performance.

Phase 3 entailed running the system on various existing designs at the MITRE Corporation, to see how these systems behave under test, and further assess the

TVLA system.

3.2 Phase 1: Design and Implementation

3.2.1 Fixed vs. Random

In various works, a general test to catch potential information leakage was a fixed vs. random implementation of TVLA. This test was implemented in the MITRE side-channel framework, through various steps. First a file in json format that contained required information on the data was created. This file contained such information as the plain text message, and meta data information. Meta data information included how many traces there are in total, the key, the type of test to be run, and whether a trace was a part of the fixed or random data set. Figure 3.1 shows the flow diagram for the stimulus generation file.

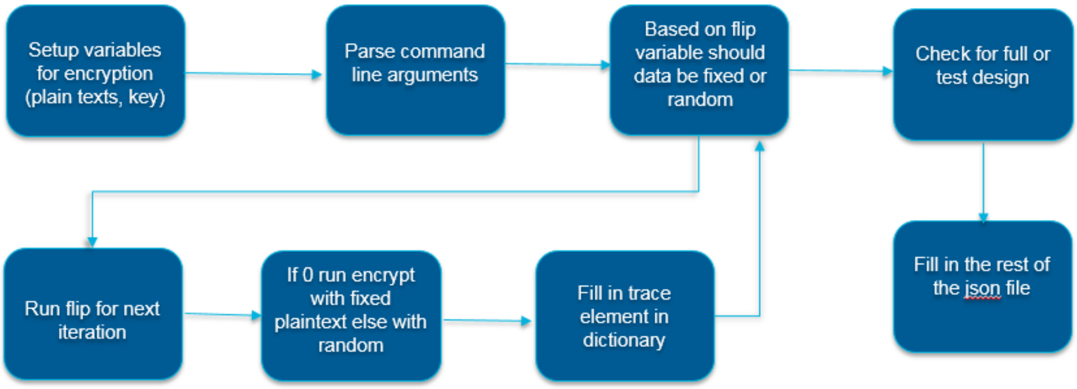
Setup variables for encryption (plain texts, key)
Parse command line arguments
Based on flip variable should data be fixed or random
Check for full or test design
Run flip for next iteration
If 0 run encrypt with fixed plaintext else with random
Fill in trace element in dictionary
Fill in the rest of the json file

Figure 3.1: Flow Diagram for Stimulus Generation

The plain text messages were designed specifically for this implementation. The first set of plain texts were chosen to be fixed and the second set of plain texts were chosen to be varying. For both datasets the key was 0x0123456789abcdef123456789abcdef0. The sets consisted of the following.

- plaintext for dataset1 (fixed): 0x00000000000000000000000000000000 (16 0 bytes) perform N encryptions where I_{j+1} = AES(K,I_j)

- plaintext for dataset2 (random): 0xda39a3ee5e6b4b0d3255bfef95601890 perform M encryptions on the same plaintext

This json file is supplied as an input into the generation of traces. Either simulations or hardware captures are run to obtain traces.

After traces are obtained, and stored they are read and processed by python code. This python code obtains the needed statistical information needed to perform TVLA. These traces then went to an analysis script which ran them through TVLA. This implementation of TVLA consists of reading in the traces, and then putting them through various processing steps. For basic TVLA the traces that are part of the fixed and random data set have there mean and variance computed separately.

After this computation they are run through Welch's t-test. There are various intermediate steps that can be computed on these traces as well, such as cropping to the attack range or adding noise into the traces. Figure 3.2 portrays a simple design for fixed vs. random TVLA.

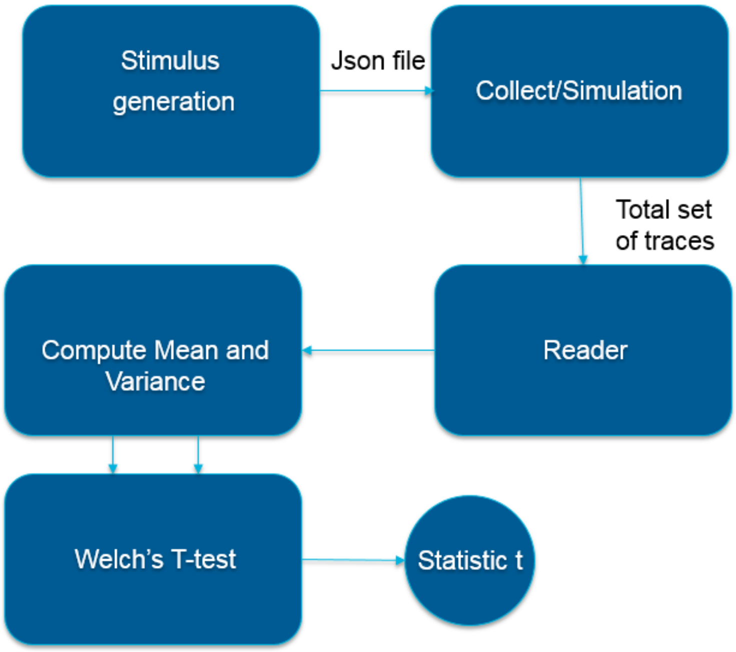

Figure 3.2: Flow Diagram for Basic Fixed vs. Random Matched Pairs

3.2.2 Matched Pairs

The matched pairs TVLA implementation is similar. In the collect step it differs in the fact that each trace is either matched with the one before or after it. Meaning that for one there is the same number of fixed and random traces and two the only thing being changed is whether the fixed trace or random trace will occur first.

An additional step is added to the analysis script. A difference step is added where the two groups are subtracted by each other, each matched pair is subtracted resulting in a difference vector. The Welch t-test step is also altered to operate on this new difference vector.

3.2.3 Higher Moment

The collection process for the higher moment implementation is identical to the matched pairs implementation. The two major parts that change are the analysis script and the difference step. The analysis script must now run a step for the mean before the difference in order to obtain the two means.

The difference step must now handle higher orders through differencing the mean of each group from the trace and raising it to some power or performing the center product to some power before the difference of the two groups.

3.3 Phase 2: Testing and Verification

3.3.1 Unit Testing

In order to verify that the TVLA system was working as desired numerous unit tests were created. In creating these unit tests the goal was to isolate certain statistical moments using specifically crafted data sets. The first through fourth moment were tested. Statistical moments were chosen because TVLA is primarily for detecting the differences in these moments between different data sets.

The first of these data sets were to have a difference in the first moment known as the mean, and no differences in the rest of the moments. The second data set had differences in the second moment known as the standard deviation and the fourth moment. The third data set had only a difference in the third moment. The fourth data set was to only have differences in the fourth moment. The crafting of these data sets are described in full detail in the results section.

3.4 Phase 3: Application of Test Vector Leakage Assessment

3.4.1 Test Designs

TVLA was run on the traces of several test designs in order to test performance of these designs and the operation of TVLA. These designs were tested in the first through fourth moment for information leakage. In order to specify which design was to be run, the simulation and hardware capture tool, took in an argument into their corresponding scripts.

Another performance test on TVLA that was used was a comparison versus a DPA on the same data set. Matched pairs TVLA for one of the test designs was compared against the basic TVLA for the same design. The following section describes these designs.

Simple S-box

This test design entails a simple s-box look-up of a key/value pair applied twice. There is an internal register holding the input value, first s-box result, and second s-box result. The input is an 8-bit key, 8-bit data, and control signals. The second s-box result is output from the design. The inputs are supplied via a control PC, which randomly generates data with a constant key.

Masked Design

This is a quasi-protected test design. It is the initial design run through an initial t-private protection. For this protection t=1, meaning that each input bit should have one corresponding mask input bit. Random mask inputs are fed from the control PC along with the data and key.

Toggle-protected Design

This is as well the initial design, however now it is run through a second t-private protection. In an attempt to deal with leakage seen in the quasi-protected design, double the masks were used in this design. Since the t-private circuit is the same and only the mask control changed, it is essentially toggling between two sets of masks.

Chapter 4

Results

4.1 Overview

In order to determine the use and efficiency of the tool created during this MQP, multiple tests were run on the system. This section shows the results of these tests. Four main tests were focused on for this system. These tests determine the usability and effectiveness of this tool.

The first of these tests consisted of composing various data sets and observing how the t-statistic behaved at an increasing amount of traces and several moments. The second of these tests compared the output of the TVLA system with the output of a DPA being run on the same data set.

The next test was used to view how TVLA runs in terms of the several designs at MITRE. The designs under test were a simple AES s-box design, A masked design, and a toggle protected design. The various tests ran on this system showed the performance of this tool under certain circumstances.

4.2 Test Vector Leakage Assessment Unittesting

4.2.1 Overview

For this test numerous data sets were composed, which were used to test the detection capability of TVLA at several moments. The first through fourth moment were under test.

4.2.2 Crafting Datasets

Mean

The first moment which is also known as the mean was tested by having a single data set and adding an offset of 1 to the whole data set. This would keep the second through fourth moment the same resulting in only a change in the mean or 1st moment. This data set was composed by having for example all even numbers from 0 to N then adding one to that data set thus having all odd numbers from 1 to N+1. The following figure shows a generalization of how the Probability Distribution Function for these two data sets look.

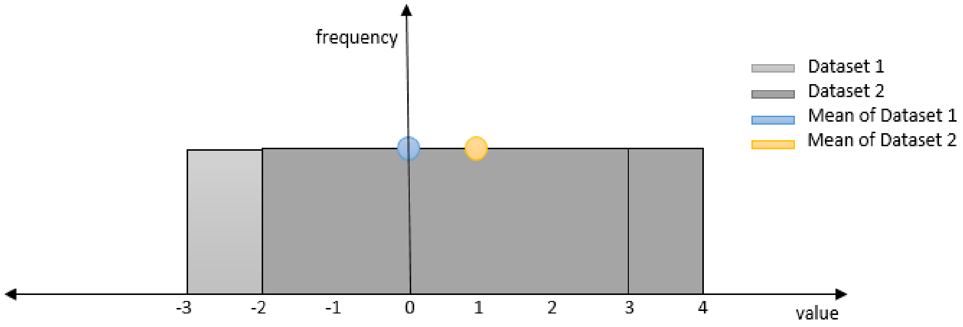

Figure 4.1: Mean Data Set

Standard Deviation

The second moment is the standard deviation. In order to test this data set a data set of zero mean and skewness was composed. This caused the first and second moment to be zero. The distribution can also be described as -N to N set, resultingly being of 2N size in total. In most practical cases if there is a difference in the second moment there will always be a difference in the fourth moment. Resultingly this test shows a difference in both moments. Figure 4.2 portrays the PDF of the data set.

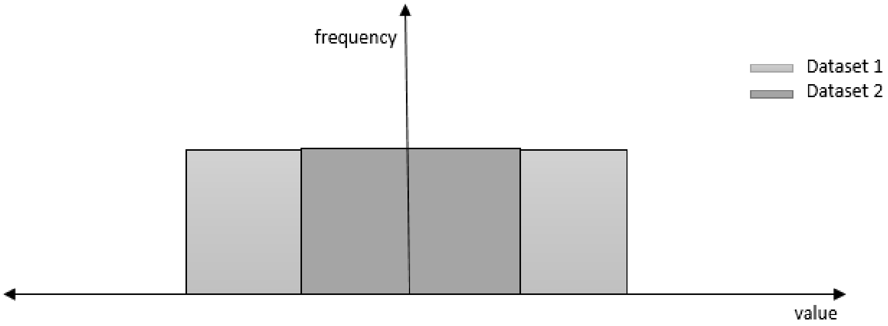

Figure 4.2: Standard Deviation Data Set

Third Moment

The third moment is used to help determine skewness. Resultingly in this data set to obtain a difference in the third moment but the same in all others. A data set with the shape of a ramp and a mean of zero was created. The second database is multiplied by -1 which causes the standard deviation and fourth moment to remain unchanged.

This data was composed by having a distribution from -N to N, then adding a value of $-N-1$. In order to maintain the zero mean and add skewness resultingly a positive values are added to offset the $-N-1$ values influence. This maintains the zero mean while adding a positive skew. In order to increase skewness further this data set is duplicated in the same vector. An ideal portrayal of this data sets PDF is shown in Figure 4.3.

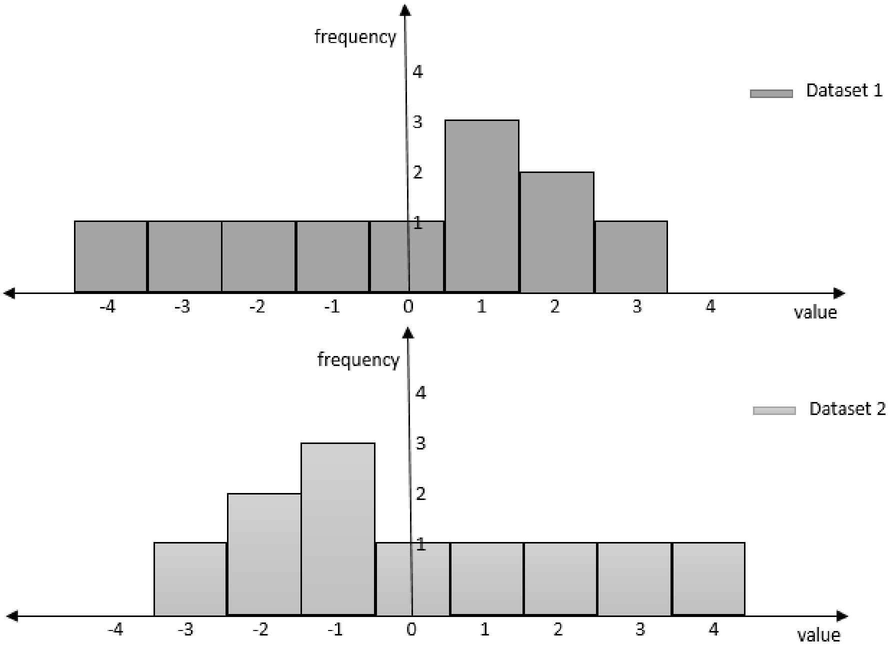

Figure 4.3: 3rd Moment Data Sets

In order to actually verify the skewness of the data set however, the PDF of the data set was plotted in python. The following figure shows this PDF and the PDF of all the values in the set multiplied by -1.

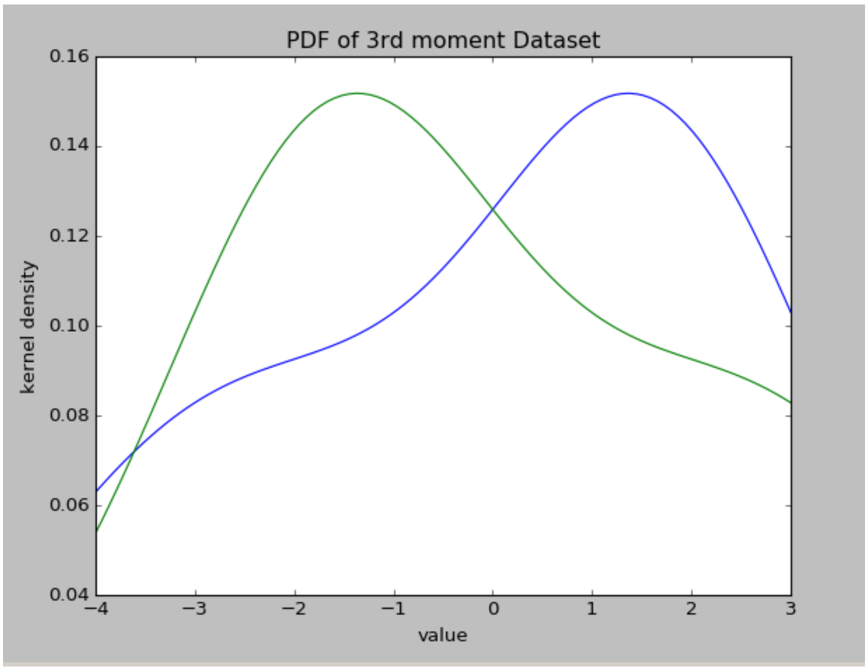

Figure 4.4: 3rd Moment Data Set PDF

Fourth Moment

The fourth moment is used to compute kurtosis. In order to obtain a data set with the same mean, standard deviation, and third moment. A zero mean, zero skewness data set was created. Following this the standard deviation of the data set is computed. The negative and positive standard deviation is then repeated to the size of the first data set creating a data set with only noticeable differences in the fourth moment. Figure 4.5 portrays these data sets.

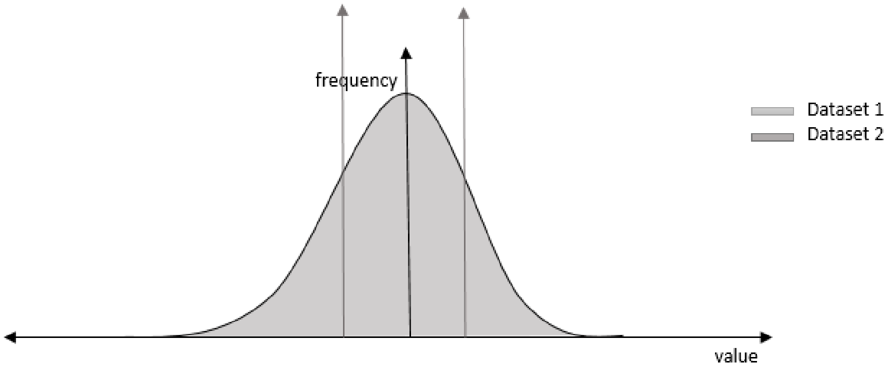

Figure 4.5: 4th Moment Data Set

4.2.3 Unit Test Results

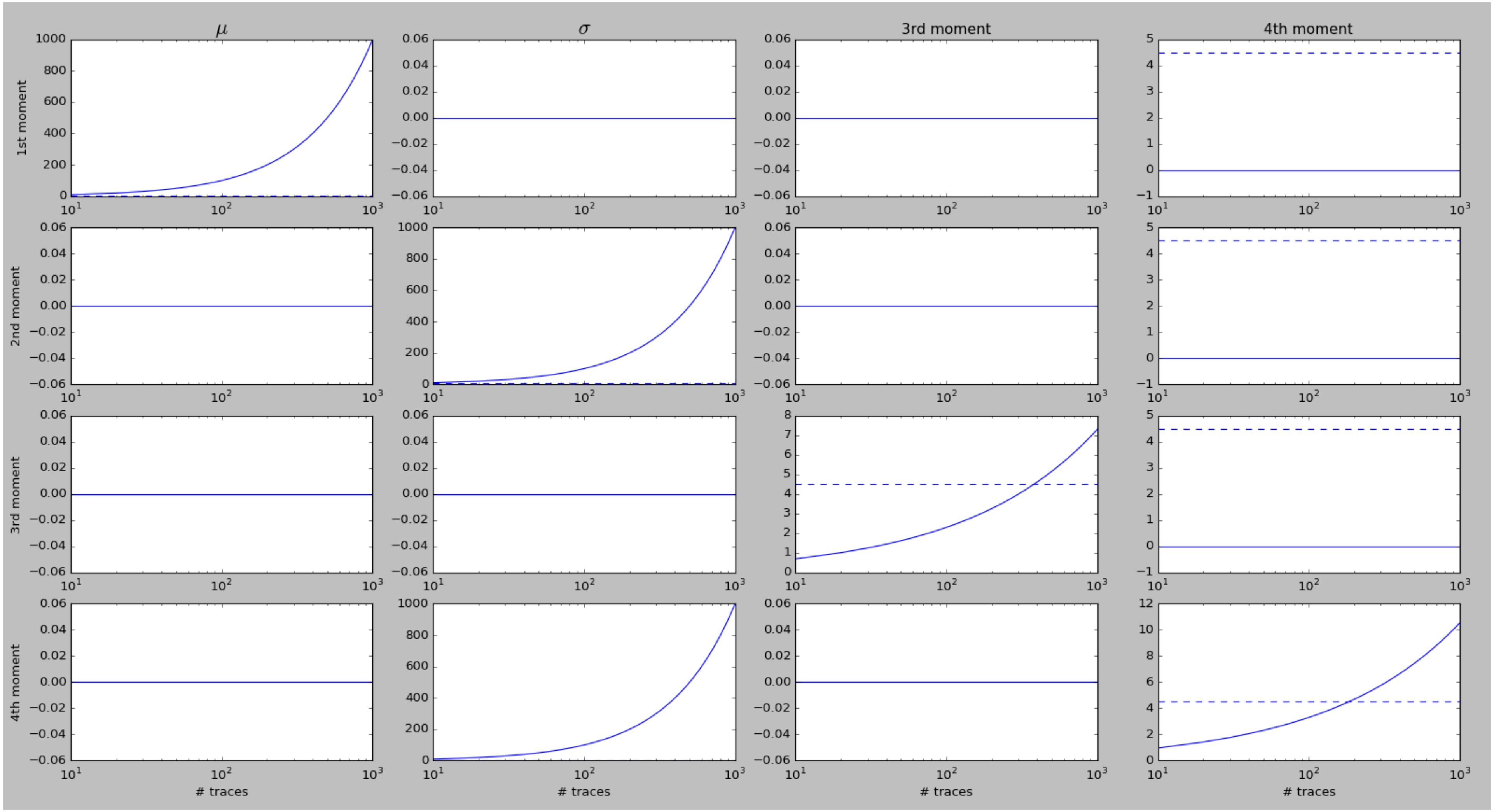

Figure 4.6: T-statistic Result for Unit Tests at first-fourth order

Figure 4.6 shows the t-statistic result along the first through fourth moment for the 4 different data sets. Along the diagonal of the figure a difference in the data sets are detected for the desired moment.

As stated before the second data set also has differences in the fourth moment and that is seen as well. The plots along the diagonal of the figure as well all rise above the 4.5 confidence value of the t-statistic.

TVLA accurately detects a difference at a particular moment for the data set. This means that it is able to see if two subsets are from the same distribution. This test proves TVLA's ability to detect if a system is subject to side-channel analysis.

4.3 Test Vector Leakage Assessment Case Study

4.3.1 Overview

The purpose of this section is to portray the effectiveness of TVLA on detecting potential information leakage on specific designs. In this section the designs under test are a simple s-box design, a t-private masked, and a t-private toggle design. These designs were tested along the first 4 moments.

Simple S-box

The first design under test is the simple s-box implementation. The input is run through a s-box transformation. Figure 4.7 shows the t-statistic result across all 4 moments.

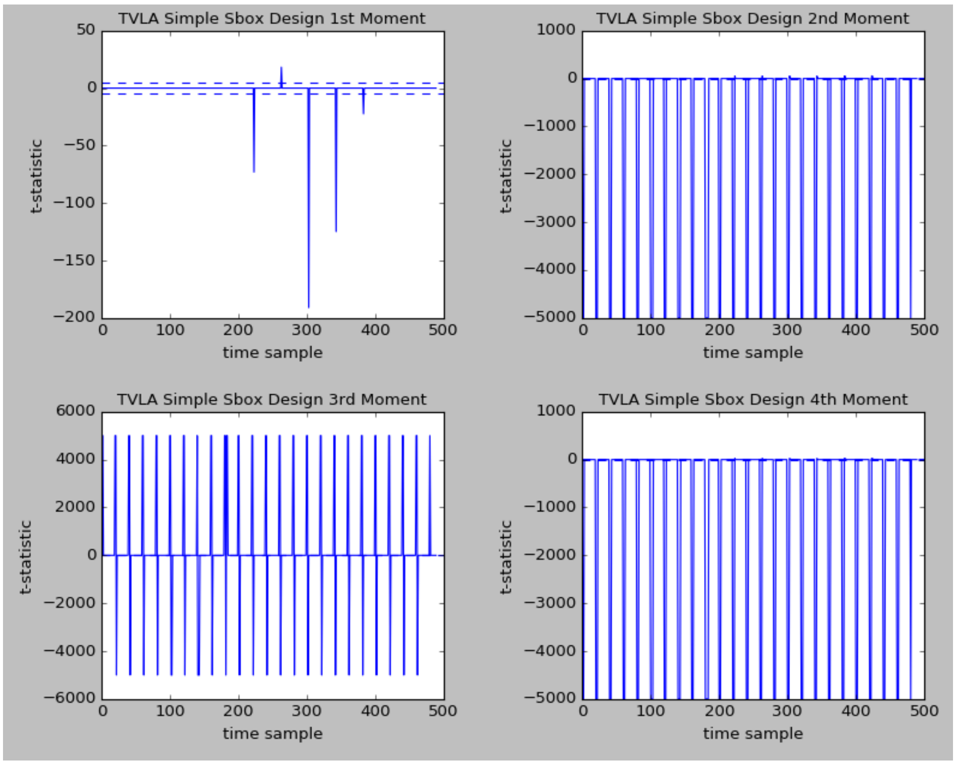

Figure 4.7: T-statistic for Simple S-box Design

As one can see and as expected there is leakage across all 4 moments, with significantly more leakage across the second-fourth moment.

T-private Masked

The second design under test is the t-private masked s-box implementation. Figure 4.8 shows the t-statistic result across all 4 moments.

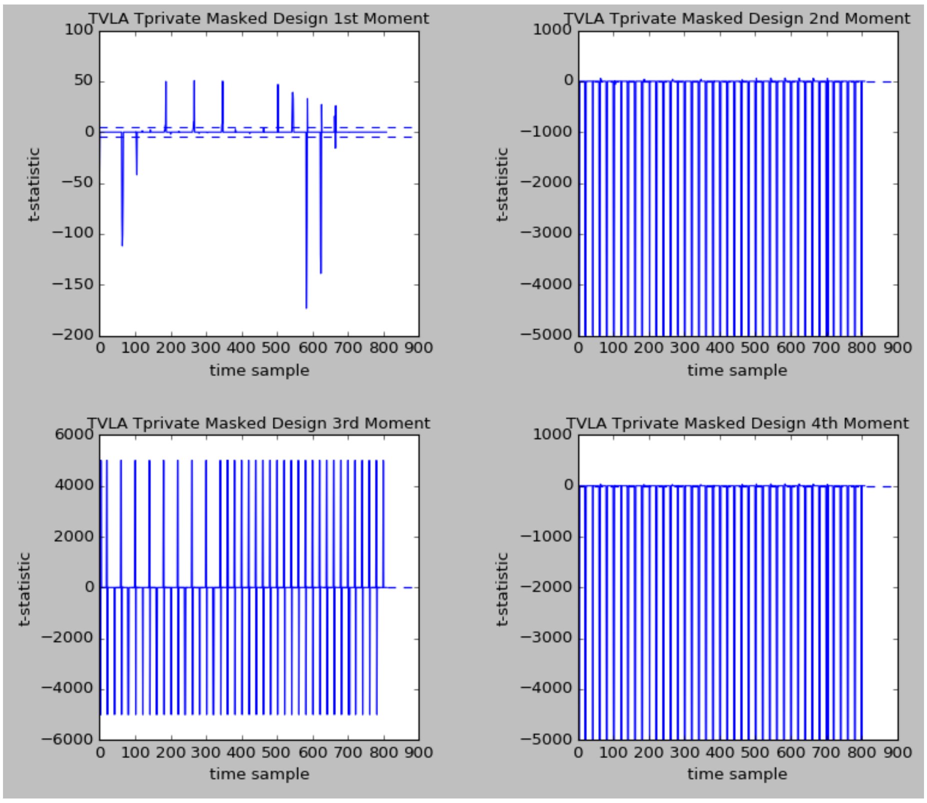

Figure 4.8: T-statistic for T-private Masked Design

This design is only quasi-protected, resultingly it still has first order leakages attributed to it. It also has no higher moment protection put into place thus the leakage at higher orders.

T-private Toggle

The third design under test is the t-private toggle s-box implementation. Figure 4.9 shows the t-statistic result across all 4 moments.

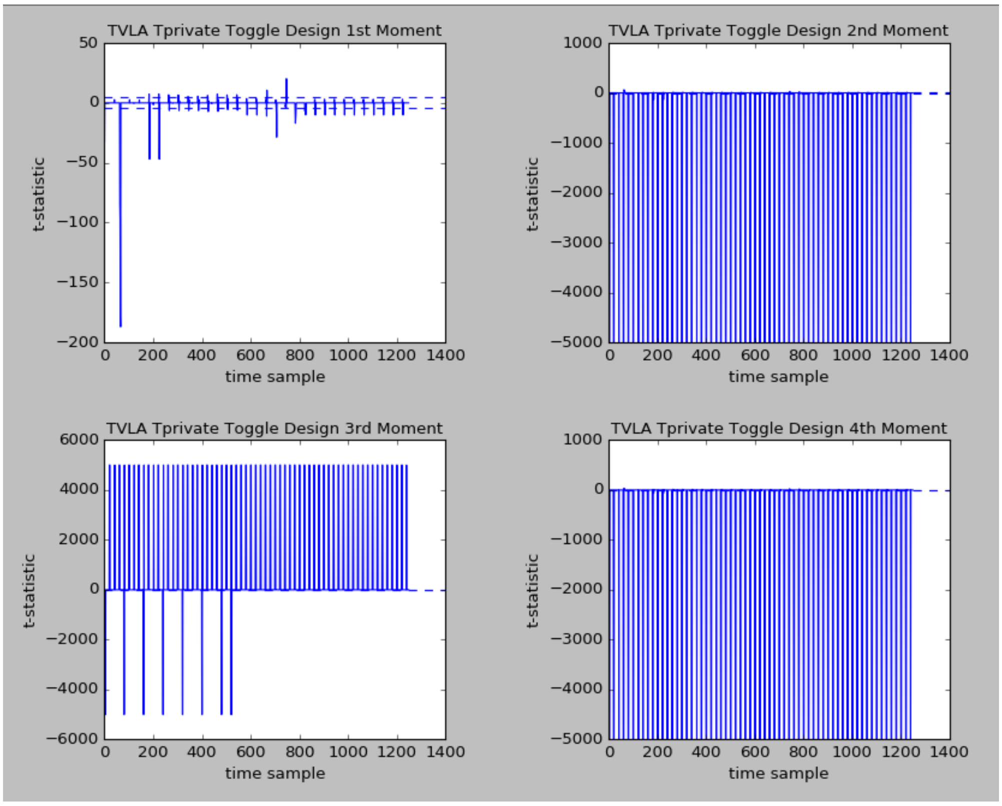

Figure 4.9: T-statistic for T-private Toggle Design

This design as well still has first order leakages attributed to it. It also has no higher moment protection put into place thus the leakage at higher orders.

4.4 Test Vector Leakage Assessment vs. Differential Power Analysis

4.4.1 Overview

In order to determine the efficiency of TVLA, the traditional DPA and TVLA were compared against each other.A simple s-box, t-private masked, and t-private toggle protected design were compared against each other. Figure 4.10 shows the comparison between these two different analysis techniques.

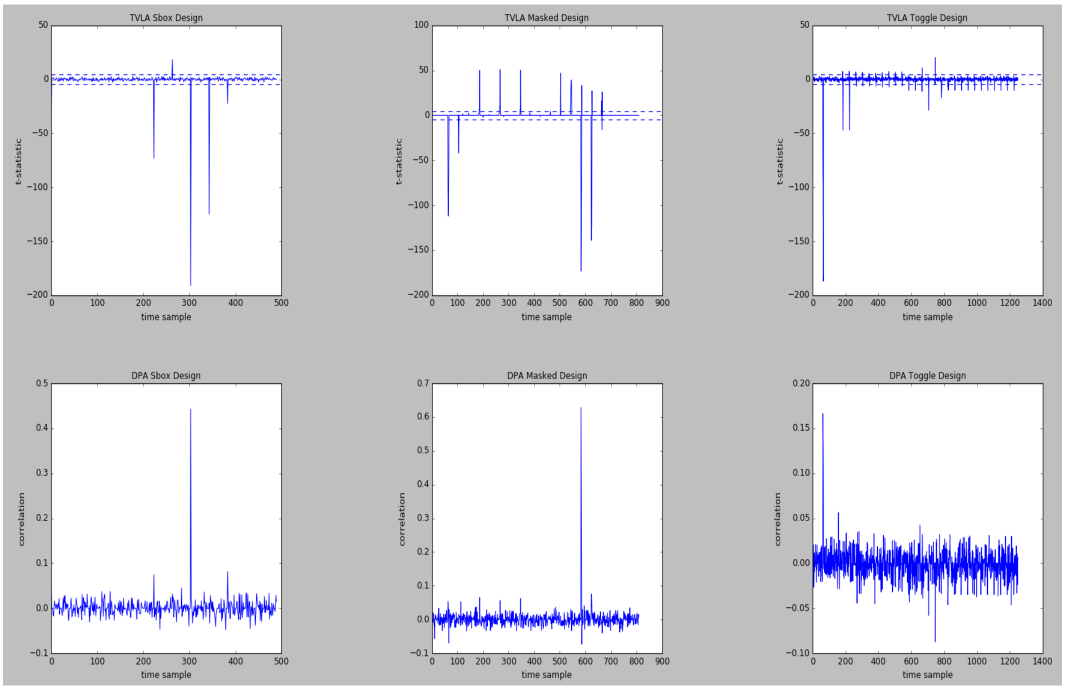

Figure 4.10: TVLA vs. DPA Results

For all three designs the TVLA shows the t-statistics beyond the 4.5 metric. The DPA as well shows a hit for all three designs. The TVLA test however, provides more information as for example in the TVLA Toggle Design it seems as though the TVLA is hinting at there being a bit of information leakage at each toggle of a mask. Another improvement this TVLA system provides is higher moment leakage detection as shown in the previous case study section. As seen above currently all designs show higher moment information leakage.

4.5 Basic vs. Matched Pairs Test Vector Leakage Assessment

4.5.1 Description

This test was done in order to show the performance of the matched pairs TVLA under environmental fluctuations compared to the basic TVLA. In order to perform this test a script that ran the basic TVLA and matched pairs TVLA implementation at various noise levels was created. This resulted in numerous vectors representing t-statistic values. However, the only concern was seeing the difference in performance between these two systems. The traces were reduced to a single point in time where leakage was known to occur. In this case it was during the second clock cycle.

A gauge of noise in the lab was desired. In order to accomplish this task a significance value was derived. Through this significance value the change due to environmental fluctuations could be obtained.

4.5.2 Deriving Significance

In determining the lab significance. The equation below was followed.

$$Significance = \frac{|\mu_a - \mu_p|}{\sigma_{lab}} \tag{4.1}$$

This significance value was calculated for the lab, and used to obtain the noise level. A basic temperature model was assumed for the cause of the environmental fluctuation. The μ_a and μ_p values represent the varying horizontal means of the environmental fluctuations across all of the traces. The sigma represent the standard deviation at a point in time where leakage is known to occur. The second clock cycle was used.

Environmental Fluctuations

The noise level in the lab is based on an approximation of this significance value. The below equation shows a complete representation of this environmental fluctuation.

$$Env.Fluctuations = \frac{Significance * \sigma_{sim}}{2} \tag{4.2}$$

The standard deviation at a point of leakage for the simulation is used in order to calculate what the difference between the means would be for the DC offset of simulations should be.This difference is then halved.

Mean

The horizontal mean across all traces was determined and plotted it in order to determine the environmental fluctuations for each particular trace. This type of averaging can be seen as a low pass filter thus obtaining the DC Offset of each trace.The following figure shows the horizontal average for all traces. Figure 4.11 shows this DC Offset for an unprotected s-box implementation for 200,000 traces.

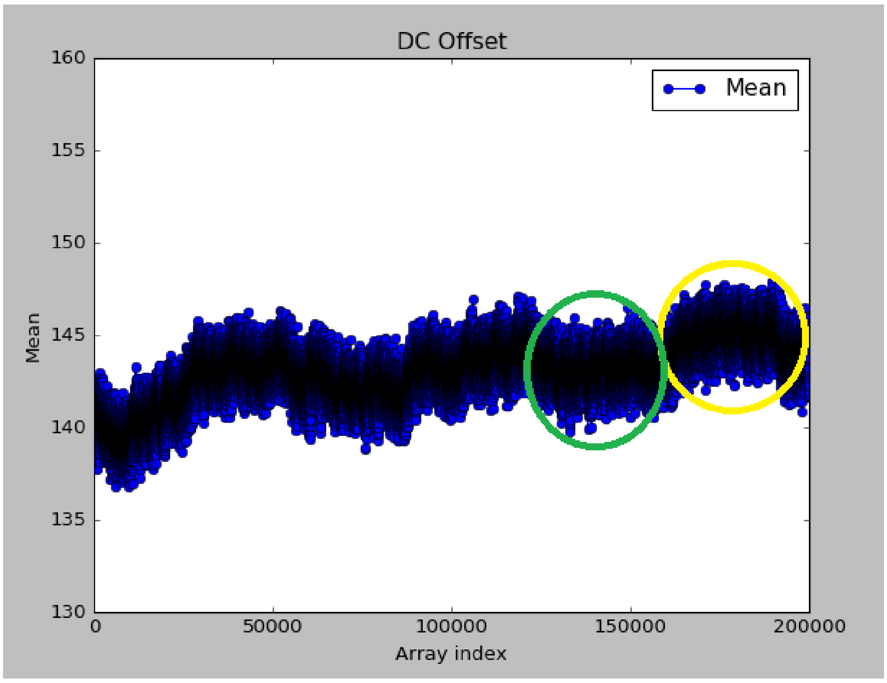

Figure 4.11: DC Offset for 200,000 Traces

Since the environmental fluctuations seem sinusoidal across all traces, a temperature model was adapted for the environmental fluctuations. The means of the environmental fluctuations during the day and the night was desired in order to calculate how significant lab noise was. A 20,000 trace range was chosen for obtaining these means. In Figure 4.11 the yellow circle represents the subset of DC offsets used when obtaining the day mean and the green circle represents the subsets of DC offsets used when obtaining the night mean.

Standard Deviation

In calculating the lab standard deviation, a time of leakage must be chosen for the set of traces. A reference trace was plotted to determine this time. Figure 4.12 shows this plot.

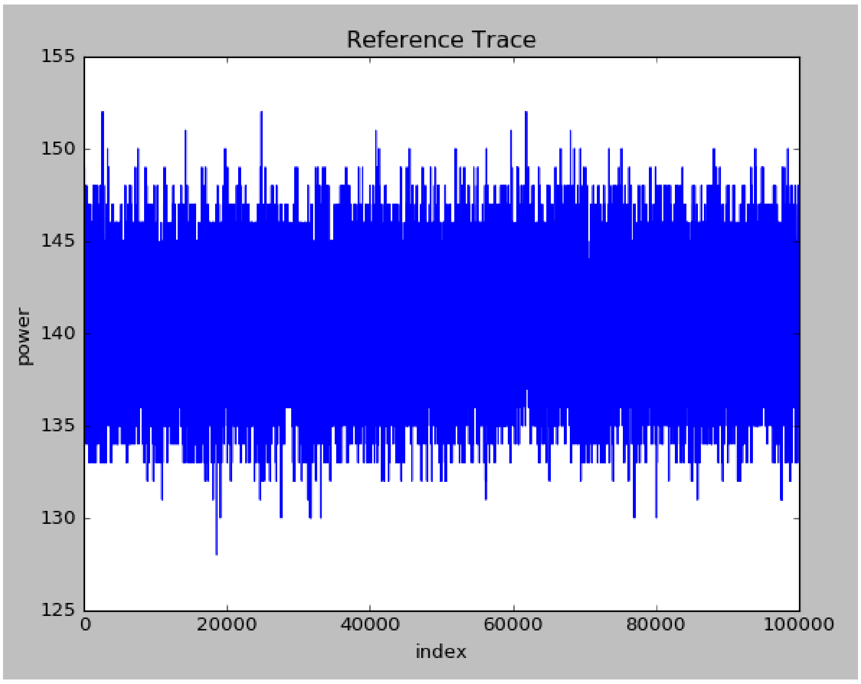

Figure 4.12: Reference Trace

When viewing this trace it is incredibly difficult to detect where the second clock cycle may be. In order to make viewing easier the mean across all traces was computed, which resulted in a much clearer image. The window size was cropped to the corresponding attack window. Figure 4.13 shows the plotted average vector. The vertical line corresponds to the second clock cycle.

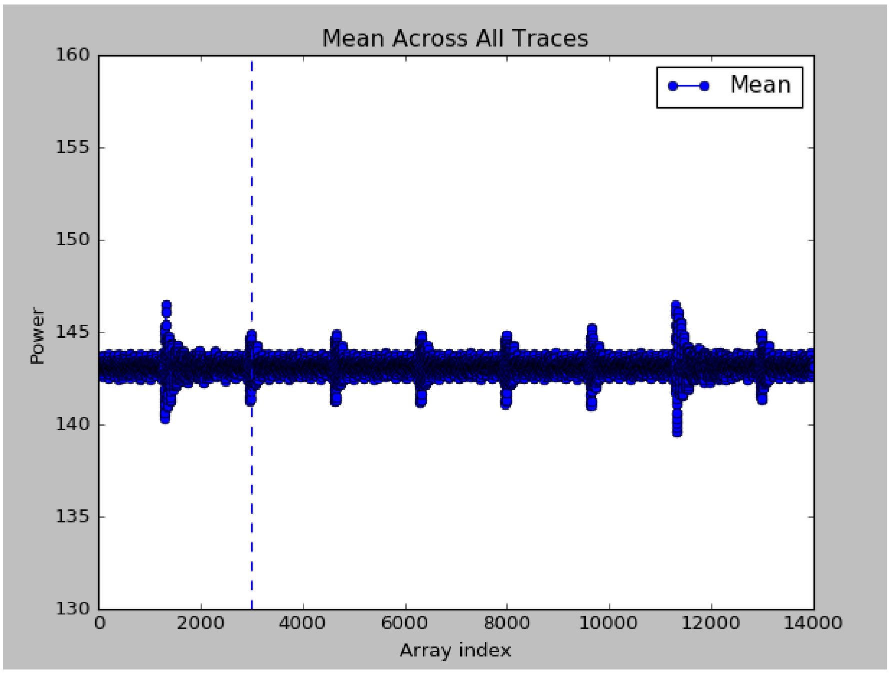

Figure 4.13: Averaged Reference Trace over entire Set of Traces

The standard deviation across all the traces at this point is computed. The same is done for simulation data. However, a reference trace in simulation showed a much clearer picture of where the second clock cycle is. Below is a plot of a reference trace. The second clock cycle is seen to be around array index 260.

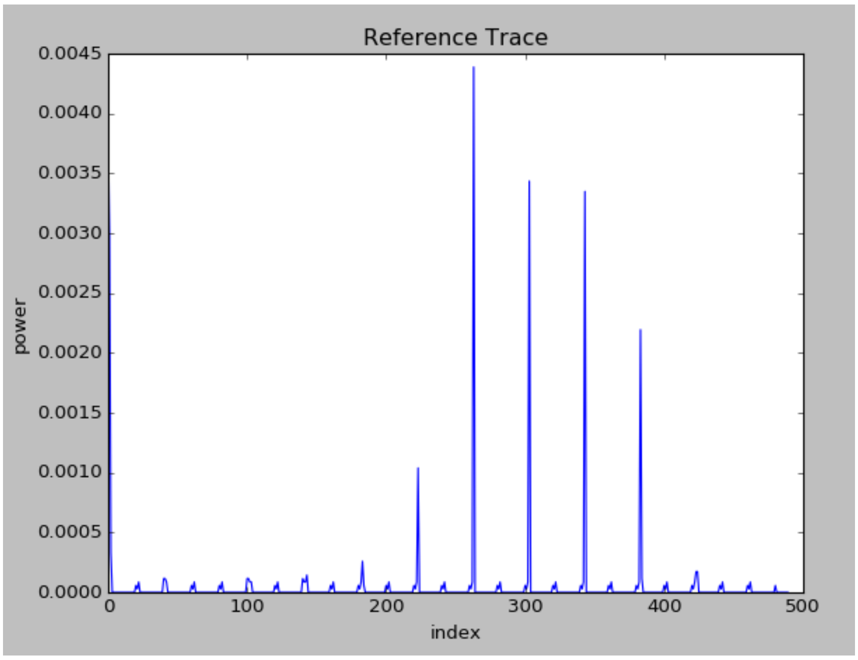

Figure 4.14: Simulated Reference Trace

4.5.3 Generating T-statistic

In order to generate the t-statistic environmental fluctuations resembling that in the lab were added. The simulations would only be 10,000 traces compared to the lab captures which were 200,000 traces. A different DC Offset was added to each trace and made it so that the offset changed in relation to a sinusoid with a similar period as the sinusoid shown in the DC offset graph. The DC offset resultingly changed slowly over the number of measurements. The following graph is a plot of the dc offset of these simulated traces at various noise values.

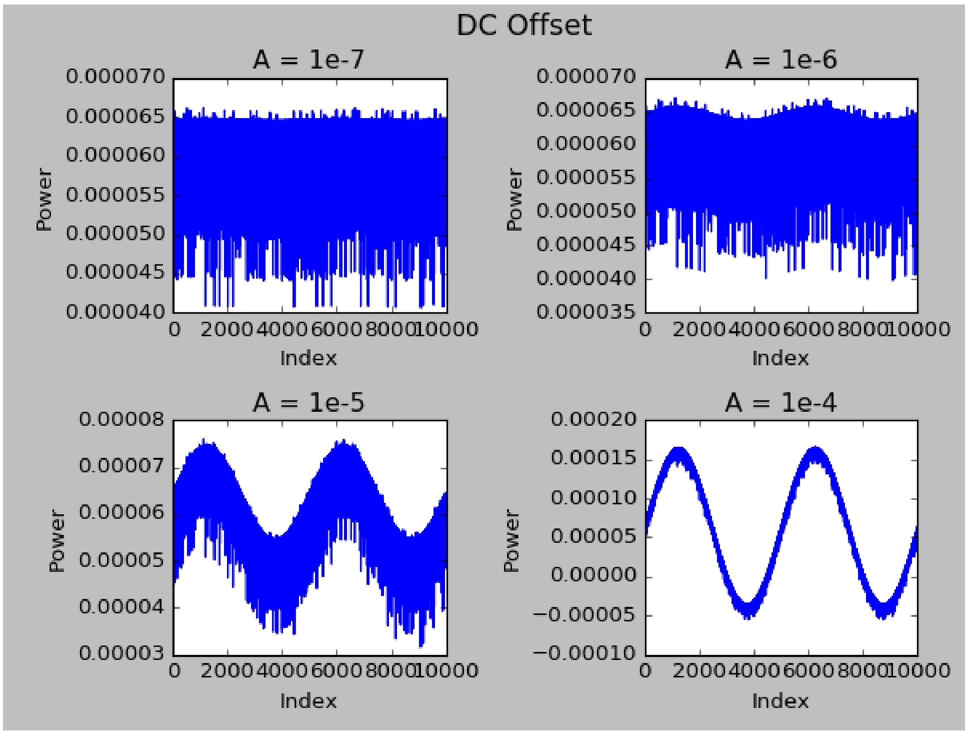

Figure 4.15: Horizontal Mean at Various Fluctuations

The range decided on consisted of $A = 1E - 7$ to 1. A is representative of the magnitude of change. It is simply a multiplier or the amplitude of the added sinusoid.Noise was added to both the matched pairs and the basic TVLA implementation. Figure 4.15 shows the detection performance of these systems at various noise levels.

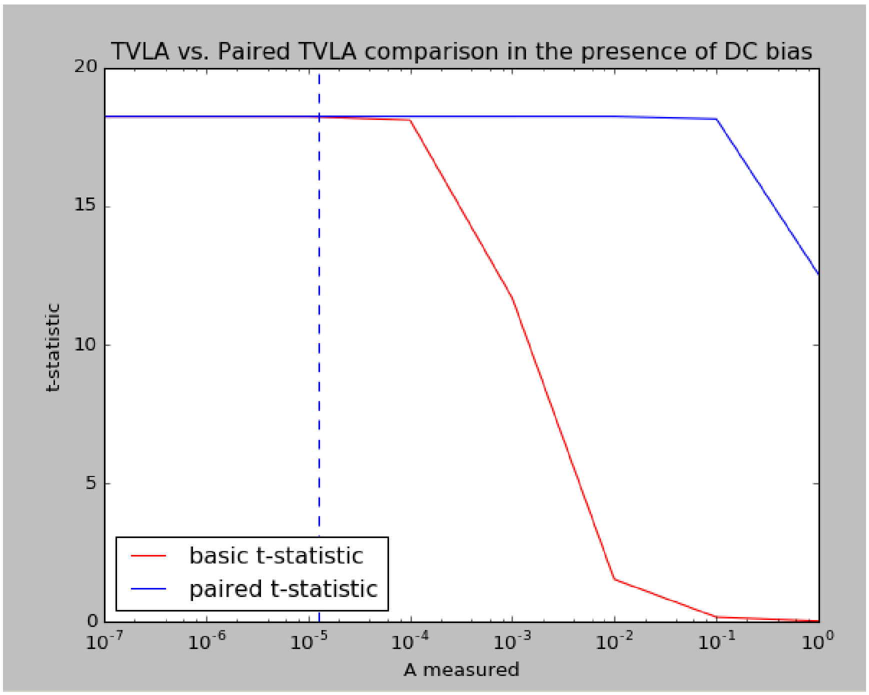

Figure 4.16: T-statistic of Data set at the second clock cycle

Figure 4.16 shows the t-statistic at the second clock cycle. A clock cycle where leakage is known to occur. As one can see the paired t-statistic in blue is able to detect leakage at higher noise levels showing its resistance to greater noise levels. The dashed vertical line shows the approximated value for lab environmental change. The difference is about 3 orders of magnitude which proves the matched pairs implementation increased ability over the basic TVLA implementation. One resultingly sees that as well a strong noise would be needed for the matched pairs TVLA implementation to realize its full potential.

Chapter 5

Conclusion

5.1 Test Vector Leakage Assessment Tool

The goal of this MQP was to develop a Test Vector Leakage Assessment system that allowed an individual to test a device and or its countermeasures for efficiency. This system needed to be able to function within the MITRE side-channel framework. The design and implementation needed to be well documented and reviewed thoroughly by MITRE staff. This was achieved through presentations on the code architecture, algorithms used. The tool was successfully constructed and tested thoroughly for performance and efficiency.

5.2 Outcomes

Through this tool one is able to see if ones system is leaking information. In order to make sure the tool is functioning properly numerous tests were run on it and the results of these tests were documented. The first test ran was in order to see if the tool was able to detect differences between distributions with different statistical moments.

If it was able to detect this difference then it could detect this difference in two differing datasets that should have the same or close distributions during an encryption run on a device, but differ due to the effect of input data and the key on power consumption.

The second of these tests were on specific designs, which were a simple s-box design, a t-private masked design, and a t-private toggle design. These designs were run through the TVLA tool at the 1st through 4th moments, and outputted a t-statistic, which shows whether or not information leakage is occurring.

The third of these tests compared a DPA output against the TVLA output at the first moment. This comparison was in order to view the performance of TVLA against the traditional DPA.

The fourth test compares TVLA against a matched pairs implementation. This comparison is done as the matched pairs implementation is able to show leakage on a system despite great amounts of environmental fluctuations. The results show that in a first moment TVLA test matched pairs provides a three order of magnitude improvement over the base TVLA, when noise is introduced into the system.

5.3 Recommendations

Over the course of this project various concerns in the flow of the system that could use improvement were identified. This section documents a few of these issues.

5.3.1 Memory Management

Memory I/O

A reoccurring issue was with memory I/O. Often times when storing large quantities of data a t-statistic value can produce an erroneous value. Producing an erroneous value is extremely troubling as it may say that your system is safe even though, it is vulnerable to devastating attacks. Thankfully, this error can be prevented by clearing directories between runs of the system and spreading out data in several directories, but this issue should be dealt with in the future.

Fragmentation of Data

This issue derives from the fact that currently data is all stored in a given directory. However, directories have a max amount of files that can be stored in them, and on long runs these directories can reach extremely large sizes, causing access slow downs. If possible the system should enforce separtion of data into several directories.

5.3.2 Cache Detection

Currently if there is a change to a functional block known as a "step", and the system is rerun. The previously cached value will be retrieved even though the step might process the data in a new way. If possible the system should detect that these steps have changed and rerun the system instead of retrieving outdated cached values.

5.3.3 Profiling

Often times, the system takes much time in simply starting. The system if run numerous times in a single script may take a while to run. The system also currently does not have any profiling established into it. One does not know where time is being spent. It would be great to add profiling data into the system to be able to see this in potentially mitigate long run time.

Acknowledgments

This project was made possible through support of individuals at The MITRE Corporation and Worcester Polytechnic Institute, notably Joseph Chapman, Thomas Eisenbarth, and Daniel Sirgey. Thank you for your outstanding help during this project.

Bibliography

[1] "A brief history of cryptography," https://access.redhat.com/blogs/766093/posts/1976023, 2013.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," http://link.springer.com/chapter/10.1007/3-540-48405-1_25#page-1, 1999.

[3] "Cracking sim cards with side-channel attacks," http://www.rambusblog.com/2015/08/11/cracking-sim-cards-with-side-channel-attacks/, 2015.

[4] G. Goodwill, B. Jun, and J. Jaffe, "A testing methodology for side-channel resistance validation," 2011. [Online]. Available: https://42xtjqm0qj0382ac91ye9exr-wpengine.netdna-ssl.com/wp-content/uploads/2015/08/a-testing-methodology-for-side-channel-resistance-validation.pdf

[5] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, 1st ed. Taylor and Francis Group, 1996.

[6] M. Whitman and H. Mattord, *Principles of Information Security*, 4th ed. Course Technology, 2011.

[7] T. Damico, "A brief history of cryptography," http://www.inquiriesjournal.com/articles/41/a-brief-history-of-cryptography, 2015.

[8] "The black chamber," 2016. [Online]. Available: http://simonsingh.net/The_Black_Chamber/maryqueenofscots.html

[9] "Symmetric cryptography," 2016. [Online]. Available: http://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.13/gtps7/s7symm.html

[10] M. Bellare, "Symmetric encryption," 2015. [Online]. Available: https://cseweb.ucsd.edu/~mihir/cse207/w-se.pdf

[11] "Cryptography competitions," https://competitions.cr.yp.to/aes.html, 2014.

[12] R. Weaver, D. Weaver, and D. Farwood, *Guide to Network Defense and Countermeasures*, 3rd ed. Course Technology, 2013.

[13] "Asymmetric algorithms," http://www.encryptionanddecryption.com/algorithms/asymmetric_algorithms.html, 2016.

[14] H. Bar-El, "Introduction to side-channel attacks," 2010. [Online]. Available: http://gauss.ececs.uc.edu/Courses/c653/lectures/SideC/intro.pdf

[15] A. Do, S. T. Ko, and A. T. Htet, "Introduction to side-channel attacks," 2013. [Online]. Available: https://web.wpi.edu/Images/CMS/ECE/MQP_Report_EM_Analysis__6.pdf

[16] "William sealy gosset," 2003. [Online]. Available: http://www-history.mcs.st-andrews.ac.uk/Biographies/Gosset.html

[17] A. Ding, C. Chen, and T. Eisenbarth, "Simpler faster and more robust t-test based leakage detection," 2016.

[18] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Revealing the Secrets of Smart Cards*, 1st ed. Springer Science, 2008.

[19] M. Hazewinkel, "Moment," 2001. [Online]. Available: https://www.encyclopediaofmath.org/index.php/Moment

[20] Y. Ishai, A. Sahai, and D. Wagner, "Advances in cryptology-crypto 2003," 2015.

[21] Z. Goddard, T. Eisenbarth, and N. LaJuenesse, "Power analysis of the t-private logic style for fpgas," 2015.