

Mutual Information Analysis in Power Side-Channel Risk Assessment

A Major Qualifying Project: Completed as a collaborative effort between Worcester Polytechnic Institute (WPI) and The MITRE Corporation. Submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment to the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering.

Written by: Michael Paquette

Submitted: October 13, 2017

Advised by: Professor Andrew Clark (WPI), Joseph Chapman (MITRE)



Abstract

It is becoming increasingly important to secure embedded systems from physical attacks that seek to extract sensitive information. The integrity of electronic devices, with and without countermeasures, are threatened by side-channel attacks. Yet, a standardized procedure for assessing a system's information leakage is a problem that has not been fully solved. The TVLA methodology, based on Welch's t-test, is a commonly utilized tool. However, under certain conditions this test may not accurately indicate the security level of an implemented design. A more universal analysis may be desirable in some cases. This MQP will explore the benefits and drawbacks of mutual information applied to power side-channel leakage.

Table of Contents

Abstract	3
Table of Contents	4
List of Figures	6
1 Introduction	8
2 Background.....	10
2.1 TVLA and Welch’s t-test	10
2.2 Matched Pairs t-test.....	12
2.3 Higher Moment t-test	12
2.4 Linear Correlation Coefficient	13
2.5 Rank Correlation Coefficient	14
2.6 Mutual Information	16
2.7 Applied to Side-Channel Analysis	17
2.8 T-Private Masking Countermeasure.....	18
3 Methodology.....	20
3.1 Overview	20
3.2 Design and Implementation	21
3.2.1 Design Overview	21
3.2.2 Uniform Random Sampling.....	23
3.2.3 Histogram Construction	24
3.2.4 Mutual Information.....	25
3.2.5 Rank Correlation.....	26
3.2.6 Threshold Detection.....	26
3.2.6 Hardware Power Trace Collection.....	27
3.3 Testing and Verification.....	29
3.3.1 Unit Testing	29
3.4 Designs Under Test	32
3.4.1 AES Substitution Box	32
3.4.2 AES Encryption	33
4 Results and Discussion	35
4.1 Unit Testing Results	35

4.2	Experiment Results	41
4.2.1	Substitution Box.....	41
4.2.2	t-private Masked Substitution-Box	42
4.2.3	AES Encryption (Hamming Distance Simulation)	44
4.2.3	AES Encryption (Power Traces).....	48
5	Conclusions.....	53
	References.....	54

List of Figures

Figure 2.1: General approach to side-channel risk assessment.....	10
Figure 2.2: Probability distributions showing the division of a single Gaussian population	11
Figure 2.3: Several pairs of PDFs that each differ in a different statistical moment.....	12
Figure 2.4: Three scatter plots showing a linear trend.....	14
Figure 2.5: Scatter plot illustrating the difference between linear and rank correlation.....	14
Figure 2.6: t-private transformation for FPGA implementations.	19
Figure 3.1: Example stimulus file for 8-bit encryption with 1,500 traces	21
Figure 3.2: Overall system block diagram showing trace collection and final analysis.....	22
Figure 3.3: Diagram showing uniform random sample generation	23
Figure 3.4: General algorithm for calculating mutual information.....	25
Figure 3.5: Categorical rank correlation process	26
Figure 3.6: Lab setup for collecting power traces on the SAKURA GIII board.	27
Figure 3.7: Lab setup for power side-channel trace collection.....	28
Figure 3.8: A closer view at the instrumentation used in the lab.....	29
Figure 3.9: Histograms for the four datasets used in the moment varying unit test	30
Figure 3.10: Two distributions with identical statistical moments but different shapes	31
Figure 3.11: A simple logic diagram illustrating the s-box feedback design.	32
Figure 3.12: Post synthesis schematic for the t-private s-box feedback design.....	33
Figure 3.14: Testbench simulation for the AES design	34
Figure 4.1: Table illustrating expected results of the varying moment unit test.....	35
Figure 4.2: Experimental results of the varying moment unit test plotted over population size ..	36
Figure 4.3: Experimental results of the varying moment unit test plotted over added noise	37

Figure 4.4: Plots for the t-test, Pearson’s coefficient, and mutual information	38
Figure 4.5: Plots for the t-test, Pearson’s coefficient, and mutual information	39
Figure 4.6: Statistical analysis results from 10,000 artificial observation traces.....	40
Figure 4.7: Statistical analysis results for an unprotected s-box feedback-loop design	41
Figure 4.8: Leakage evaluation of a t-private masked substitution-box feedback design	43
Figure 4.9: AES encryption leakage assessment	44
Figure 4.10: Leakage evaluation of a t-private masked substitution-box feedback design	46
Figure 4.11: AES leakage evaluation results for an AES specific test targeting round 2.....	47
Figure 4.12: Plots showing average of 200 power traces	48
Figure 4.13: Plot showing that there at least exists first moment power leakage in this circuit...	49
Figure 4.14: Full analysis results for the same AES implementation.....	50
Figure 4.15: Results of power side-channel risk assessment with 100,000 traces.	51
Figure 4.16: Zooming in on results of power side-channel risk assessment	52

1 Introduction

Many modern cryptographic systems are implemented on programmable logic devices such as FPGAs. These devices operate using transistors, which close or open depending on the charge applied to its gate. The amount of switching activity during various operations largely determine the total power consumption of the device. Measuring a devices power consumption can provide insight into what operations are happening and can even reveal secret information. A power side-channel attack uses thousands of power traces to extract cryptographic keys and secrets.

The rise of side-channel attacks has resulted in years of research into countermeasures. Masking data, adding noise, and inserting random delays all distort power signals and increase the security level of an embedded system. The ability to test cryptographic systems and their countermeasures for information leakage is essential.

Testing for leakage is a complex problem. Performing cryptographic key attacks such as differential power analysis (DPA) or correlation power analysis (CPA) is one solution. More recently, mutual information analysis (MIA) has been studied as a potential attack. However, performing attacks is a time-consuming task since each possible key must be tested. Testing a system for leakage without choosing a specific attack is more valuable and efficient during development.

The Test Vector Leakage Assessment (TVLA) methodology, based on Welch's t-test, is a common tool in industry. However, this test is at risk of falsely passing an implementation since it only measures certain types of leakage. Even with higher moment versions of the t-test, the results may not always accurately represent the security level of an embedded system. For systems with masking countermeasures, covariance is overestimated which leads to an overstatement in security. Further, approximating higher moments generally causes amplification in noise and can lead to false positives. For these reasons, it is not reliable to assess an implementation by estimating statistical moments or by assuming an attacker's strategy [4].

A more generalized dependency test is mutual information (MI), which is based on entropy of observations rather than estimated statistical moments. MI makes little assumption about

adversarial strategy and leakage type. Although the complexity of MI is higher than that of the t-test, it provides a more universal side-channel leakage assessment [3].

The goal of this MQP is to explore the possibility of using Mutual Information as a discriminator for FPGA power side-channel analysis. It will show the benefits and drawbacks of such a test in comparison with the t-test and correlation for both power simulations and real hardware captures.

2 Background

2.1 TVLA and Welch's t-test

TVLA (Test Vector Leakage Assessment) is a statistical analysis aimed at assessing the amount of information leakage in an implemented design. TVLA requires a large set of observations, called traces, during a process implemented on an embedded system. A single power trace is a vector representing instantaneous power consumption over the course of some process such as an AES encryption. A large set of traces is divided into two groups based on a selector value associated with each trace. For example, a set of AES power traces may be divided based on the plaintext input; half of the set of traces may have been stimulated with some predetermined, fixed plaintext, while the other half was given random plaintext.

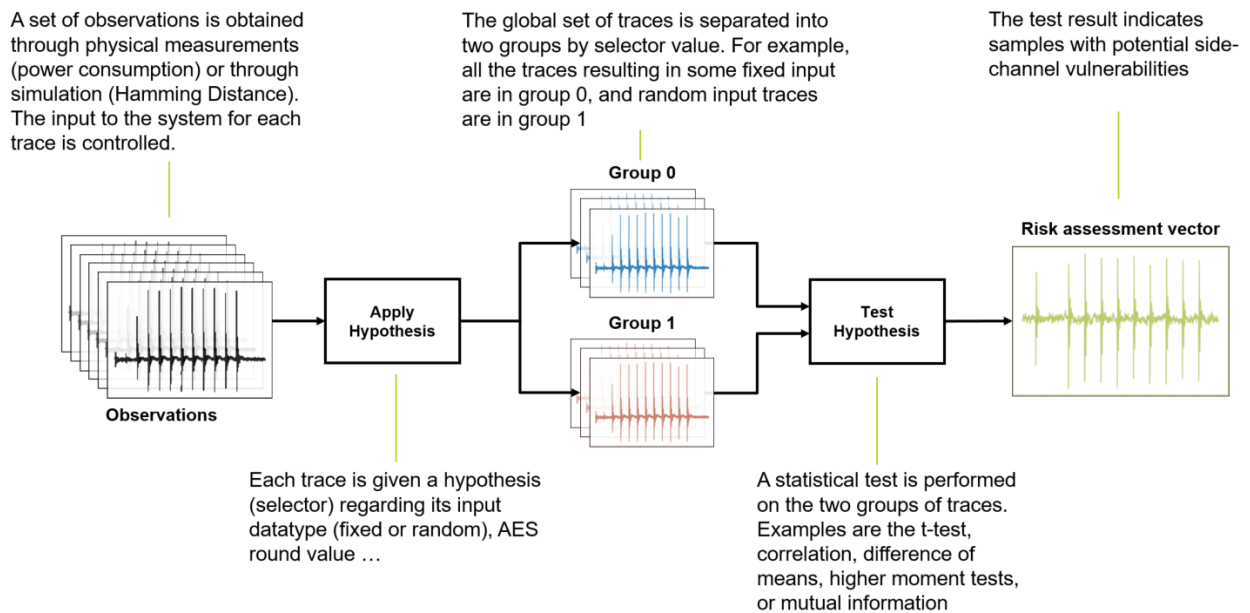


Figure 2.1: General approach to side-channel risk assessment

TVLA is based on Welch's t-test, a test with the null hypothesis that two populations of power traces have equal means at any given time sample. The null hypothesis is proven false if there is enough information available to conclude that a selector has some impact on power consumption.

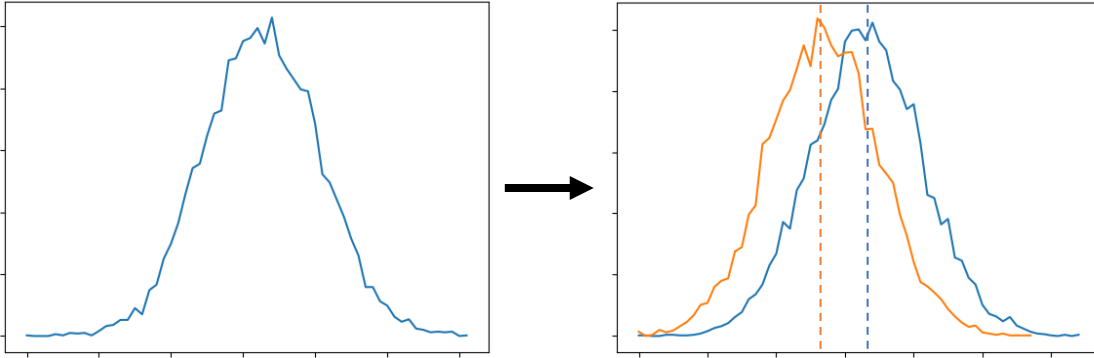


Figure 2.2: Probability distributions showing the division of a single Gaussian population (left) into two groups (right). When split, the two resulting Gaussian groups have different mean values. Therefore, the selector used to assign individual data points to each group does have some influence on the data itself.

If dependency exists, there is the potential for a vulnerability in the implementation. The statistic value is computed as follows, where \bar{X}_n is the mean of the nth population of power traces, s_n is the standard deviation of the nth population of power traces, and N_n is the number of traces in each population [2].

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \quad (2.1)$$

If the absolute value of the t statistic is greater than the threshold of 4.5, the test fails. Less than 4.5 does not necessarily mean there is no leakage; it may simply mean that the population size is not large enough [2].

There are two types of t-tests: general and specific. The general test divides the set of traces based on the input to a cryptographic system, which is either some predetermined fixed input or any other random value. For example, half of the traces may be associated with the fixed input 8'hDA, while the other half is associated with a random set of 8-bit values. If power is dependent on input data, the test should yield a high t-value. It is important to remember, however, that a general test cannot guarantee an attack is possible even if leakage is present [2].

A specific test only differs in the selector. While the general test uses fixed or random input as a selector, the specific test uses intermediate cryptographic values such as encryption round outputs

or s-box outputs. These are common values exploited for key extraction. Each trace in the set is associated with a random input, unlike the fixed vs. random general test [2].

2.2 Matched Pairs t-test

The matched pairs TVLA test is a modified version of the original TVLA procedure. The purpose of this variant is to reduce the effect that environmental fluctuation has on the mean and variance of the two sets. Fluctuations in ambient conditions such as temperature can make a system appear secure when leakage is present. To perform the matched pairs variant, the set of traces must undergo a step in which each trace from the first set is paired with an adjacent trace from the second set. The difference of each pair is computed and the modified t-test is performed on the difference set per the following equation [1]:

$$t_d = \frac{\bar{D}}{\sqrt{\frac{s^2}{N}}} \quad (2.2)$$

2.3 Higher Moment t-test

A distribution can be defined by an infinite set of moments. The first moment is the mean of the dataset. All remaining central moments are centered on this mean. Variance is the second central moment; it roughly describes the width of a probability distribution function (PDF) surrounding the mean. Third and fourth moments are known as skew and kurtosis respectively. Skew is a measure of asymmetry and kurtosis is a measure of tail weight. There is an infinite set of moments to a distribution, but these four are the most useful in practice [1].

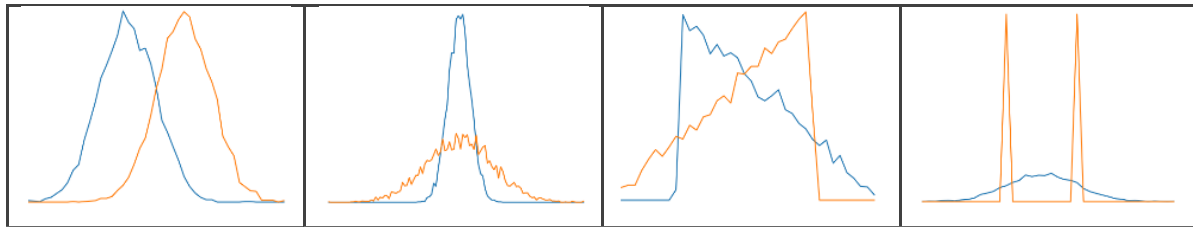


Figure 2.3: Several pairs of PDFs that each differ in a different statistical moment. The first pair of distributions have different mean values, but the same variance, skew, etc. The 2nd pair have the same mean, but different variance. The 3rd pair differs in skew, and the 4th kurtosis.

There is a higher moment implementation of the t-test that can identify higher moment leakage; it is like the matched pairs implementation, but some pre-processing must be done before the t-test step. The differencing step subtracts the mean of each group from each trace and raises that difference to the desired order. Then the actual matched pairs differencing is performed. The higher moment difference is computed as follows [1]:

$$D = [(L_A - \bar{L}_A)^d - (L_B - \bar{L}_B)^d] \quad (2.3)$$

2.4 Linear Correlation Coefficient

A linear correlation coefficient is highly utilized in correlation power analysis (CPA), an attack which attempts to extract a key by correlating the results of a leakage model with actual power consumption for every possible key guess. One common leakage model is Hamming Distance, which represents the theoretical or simulated number of binary state changes in a digital system over time.

However, correlation can also be used in leakage assessment by computing Pearson's correlation coefficient between an arbitrary selector and a system's simulated or measured leakage. Pearson's coefficient is calculated by the following equation [5].

$$P_{X,Y} = \frac{cov(X,Y)}{s_X s_Y} \quad (2.4)$$

$$cov(X, Y) = \mathbb{P}(X = 1) * (\overline{Y_{X=1}} - \bar{Y}) \quad (2.5)$$

Pearson's correlation coefficient is a measure of linear dependency between two random variables. A perfect score is +/- 1 and no dependence is a score of 0. Looking at the following figure, there clearly exists a relationship between X and Y in each case. However, this relationship is not expressed by linear correlation in some of these cases.

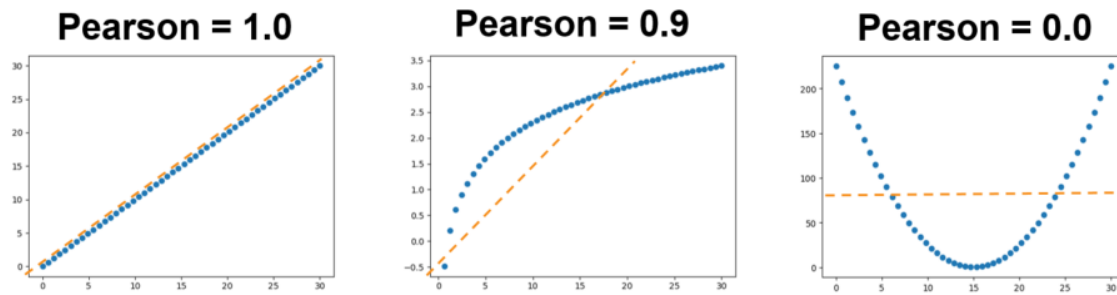


Figure 2.4: Three scatter plots showing a linear trend, a logarithmic trend, and a U-shaped trend from left to right. The linear trend scores a perfect 1.0 for Pearson’s coefficient. The logarithmic trend scores a 0.9 because there is a trend, but it is not perfectly linear. The last plot shows nonlinear correlation that cannot be expressed by Pearson’s coefficient.

In the case of binary selectors, linear correlation is essentially a comparison between the means of two groups, like the first moment t-test. They only differ in the fact that the t-test is proportional to population size. In fact, Welch’s t-test and Pearson’s correlation coefficient for a set of power traces will appear identical expect for the vertical scale.

2.5 Rank Correlation Coefficient

There are also correlation coefficients that do not only measure linear dependence. These coefficients correlate the ranks of two random variables, not the actual values. A rank is an integer value describing the sorted position of a value in a set of observations. It is therefore possible to have a perfect correlation score even where X is not related to Y by a constant factor [8]. The following figure illustrates this fact.

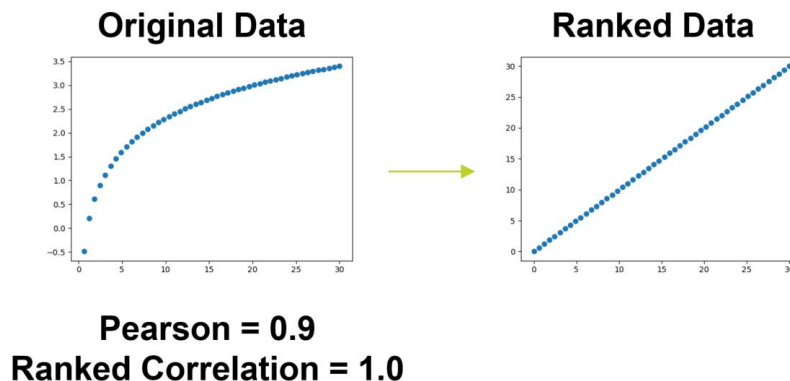


Figure 2.5: Scatter plot illustrating the difference between linear and rank correlation.

Like with Pearson's correlation coefficient, rank correlation is scored from 0 to +/- 1. There are many versions of rank correlation; some fit this application better than others.

One widely-used rank correlation coefficient is Spearman's coefficient. It is simply the Pearson's coefficient of the ranks of two random variables [8].

$$S_{X,Y} = \frac{cov(R_X, R_Y)}{s_{R_Y} s_{R_X}} \quad (2.5)$$

If all rankings of X and Y are distinct integers (there are no ties), then Spearman's coefficient can be calculated by the following formula, where n is the number of observations and d_i is the difference between the rankings of X and Y at each observation [8].

$$S_{X,Y} = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \quad (2.6)$$

Kendall's correlation coefficient is similar except that it does not consider the distance between X and Y. It merely uses the number of concordant and discordant pairs. A pair of observations is said to be concordant if the sort order of both X and Y increase between observations or both decrease between observations. On a scatter plot of X and Y, a pair is concordant if the slope of the line drawn between the two observations is positive, and discordant if the slope is negative. Kendall's Tau is calculated as follows, where N_C and N_D are the number of concordant and discordant pairs respectively [7].

$$\tau_{X,Y} = \frac{N_C - N_D}{n(n-1)/2} \quad (2.7)$$

There is a problem with using either of these coefficients in the case of a binary random variable: they do not consider ties, which are extremely abundant when considering a binary selector. In the case of a binary selector, both of these rank correlations are essentially comparing the mean rankings of two groups. In this case, Spearman and Kendall do not provide any more information than Pearson.

Goodman-Kruskal's gamma coefficient is intended for categorical data [6]. When a data set X is split into two groups A and B, gamma is calculated by the number of (A, B) pairs where X_A is greater than X_B and X_A is less than X_B .

$$\gamma_{X,Y} = \frac{N_C - N_D}{N_C + N_D} \quad (2.8)$$

This approach can yield a perfect score even though all pairs have ties. It also captures more than the difference of means between the two groups; if the means of two groups are the same, Goodman-Kruskal will yield a non-zero score if it is more common for one group to have larger data than the other. In other words, it can identify leakage where linear correlation cannot [6].

2.6 Mutual Information

Mutual information (MI) is a general dependency test between two random variables. It is a measure of the information gained about one RV when the other is known. It makes no assumptions about the nature of the leakage it will attempt to identify, unlike the t-test which assumes leakage in a specific moment or correlation, which is only a measure of linear dependence. Mutual information is similar in that the global set of traces is split into two sets by a selector. This selector can still be a fixed vs. random input or an intermediate cryptographic value, like in the TVLA general and specific tests. Conceptually, there is mutual information if the two sets of traces have unidentical PDFs. More concretely, MI is determined by the entropies of the RVs. Entropy is calculated as follows [3]:

$$H(X) = - \sum_{x \in X} \mathbb{P}_X[X = x] * \log_2(\mathbb{P}_X[X = x]) \quad (2.9)$$

Using the entropy of each RV and the joint distribution's entropy, MI can be computed.

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (2.10)$$

This states that the information between selector X and power consumption Y is equal to the sum of the entropy of the selector and the entropy of the power consumption minus the entropy of the joint distribution. There is no MI if the sum of the individual entropies is equal to the joint entropy of the two RVs. On the other hand, there is a one-to-one relationship if X uniquely determines Y. In this case, the joint entropy will equal the entropy of Y. So, the extremes of MI can be described by the function below [3]:

$$I(X; Y) = \begin{cases} 0 & ; \text{ Independence} \\ H(X) & ; \text{ Dependence} \end{cases} \quad (2.11)$$

In the case that selector X is a binary random variable with a uniform distribution, equations 5 and 6 can be simplified, as the entropy of X would equal 1.

$$I(X;Y) = 1 + H(Y) - H(X,Y) \quad (2.12)$$

$$I(X;Y) = \begin{cases} 0 & ; \text{ Independence} \\ 1 & ; \text{ Dependence} \end{cases} \quad (2.13)$$

Since mutual information requires entropy, the two random variables must be tracked in a histogram. There are methods of calculating optimal histogram bin width, which minimize error between the histogram and the actual PDF. The Freedman-Diaconis method is a commonly used method outlined below where IQR is the interquartile range (or mid-spread), and N is the total populate size [9].

$$\text{bin width} = \frac{IQR(X)}{\sqrt[3]{N}} \quad (2.14)$$

The number of bins is equal the maximum value of X minus the minimum value of X, divided by bin width. The edges are equally spaced from the minimum to the maximum value. IQR is simply the difference between the 75th percentile and the 25th percentile.

2.7 Applied to Side-Channel Analysis

Applying information theory to side-channel analysis provides has become a topic of research in recent years. Mutual information analysis has important differences from other attacks and statistical tests. It makes minimal assumptions about observed leakage; the only one being that there is a functional relationship between leaked and observed values. As an attack, mutual information analysis works similarly to correlation power analysis, where observed leakage is correlated with a leakage model's prediction. The difference is, instead of correlation, mutual information is computed between the two vectors for each possible cryptographic key guess. CPA only measures linear correlation between observations and assumed leakage, and DPA is limited to a difference of means. This makes MIA advantageous to manufacturers; it can determine the maximum amount of information leakage in an embedded system because it does not make these assumptions. If a system can withstand a MIA key attack, it should be secure against all key attacks including DPA and CPA [3].

Studies have shown some important differences between the performance of MIA, CPA, and DPA. CPA can distinguish between a correct and incorrect key guess with the fewest number of traces, closely followed by MIA, and finally DPA. However, as trace count increases, the mutual information associated with the correct key guess becomes much larger than the incorrect guesses, relative to Pearson's correlation coefficient and DPA bias. In fact, with a sufficient population of traces MIA yields the largest difference between the correct key and the next best key guess. But, MIA is far more memory intensive as it must keep histograms for every time sample. Previous work has utilized a constant 256 bins per histogram, although this could be adjusted dynamically to reduce complexity [3].

As previously stated, it is faster and more conclusive to test a design without performing a key attack. Instead of guessing every possible key, a set of traces can be divided into two groups per some selector associated with each trace and one statistical test can be performed to determine potential vulnerabilities. DPA, CPA, and MIA attacks can be replaced with Welch's t-test, correlation (linear or ranked), and mutual information for a general leakage assessment. The limitation of the t-test and correlations are the same as those of DPA and CPA: they only measure certain types of leakage. Also, they assume the adversarial strategy of statistical moment estimation. For higher moment leakage and for systems with masking countermeasures, estimating statistical moments is not adequate. Noise negatively affects high moment calculations and masking schemes simulate a high variance resulting in an overstatement of security. An information theoretic approach, however, should yield the worst-case vulnerability if PDFs can be adequately estimated by histograms [4].

2.8 T-Private Masking Countermeasure

The *t-private* masking countermeasure is meant to secure an implementation at the logic synthesis level. If an adversary has access to t nodes of observation, this countermeasure should protect the system from side-channel attacks [10]. It works by creating t copies of each logic component: look up tables and registers. For a one-private implementation there is a single copy of each component created. For each input to the component, there is a random mask bit generated. One copy performs its function on the random bits. The other copy performs its operation on the exclusive OR of the mask bits and data bits. The XOR of the two components' outputs is equal to the original output.

By performing the *t-private* transform on each component that secret information flows through, observations cannot be correlated with the secret information. The diagram below shows the transformation.

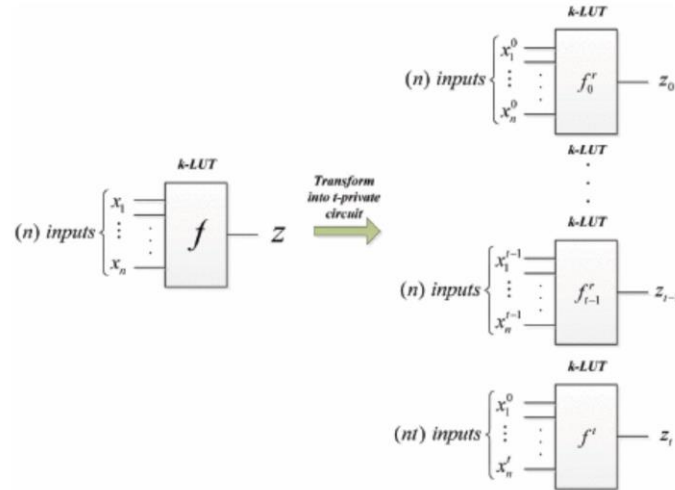


Figure 2.6: *t-private* transformation for FPGA implementations.

However, *t-private* has been proven to fail in hardware implementation. Unequal routing delays between masks and masked values cause “glitches” in cells within a single transform. These glitches cause information leakage.

3 Methodology

3.1 Overview

The tools created in this MQP output statistical vectors from a suite of statistical analyses representing information leakage of the system under test. Each type of analysis aims to locate observation-data dependency; however, each come with advantages and disadvantages under different conditions. Therefore, considering the vectors produced by multiple tests is more useful than just any one. These vectors can be used to predict potential vulnerabilities and, hopefully, assist in the design of a secure system.

Much of the project's structure was adopted from the existing MITRE side-channel analysis framework. The tools created must fit seamlessly into this existing software base. The overall design flow will be explained in detail.

The statistical analyses under consideration are Welch's t-test, higher moment t-tests, linear correlation, categorical rank correlation, and mutual information. Although some improvements were made to the t-test and linear correlation discriminators, much of this software already existed in MITRE's framework. The main contributions of this project are the mutual information and categorical correlation analyses bundled with existing statistical tests into a single program. The implementation of each statistical test will be discussed.

Extensive testing was done before analyzing any real sets of traces. Fabricated data sets were used to ensure each discriminator performs as expected, as well as to prove their predicted advantages and disadvantages. This section will also explain the hardware designs investigated by the suite of statistical analyses.

Finally, this section will go into the collection of simulated and real power traces. Collecting data for side channel leakage is not a simple task; much thought went into this subject.

3.2 Design and Implementation

3.2.1 Design Overview

The analysis tools operate on a set of trace objects, which are collected either through simulation or hardware. A Vivado project is either implemented on an FPGA or run as a post-implementation timing simulation. A JSON stimulus file dictates input data to the simulation or hardware design. This stimulus file contains the number of traces and key data common to each trace. For each trace, it contains plaintext input data and the group of traces that it belongs to (fixed vs. random). Below is an example stimulus file with 8-bit data:

```
1 {
2   "test": "fixed vs random matched pairs",
3   "traces": [
4     {
5       "plaintext": "00",
6       "datatype": "random"
7     },
8     {
9       "plaintext": "da",
10      "datatype": "fixed"
11    },
12    {
13      "plaintext": "42",
14      "datatype": "random"
15    },
16    {
17      "plaintext": "51",
18      "datatype": "random"
19    },
20    {
21      "plaintext": "da",
22      "datatype": "fixed"
23    },
24    ...|
25  ],
26  "key": "01",
27  "num_traces": 1500
28 }
```

Figure 3.1: Example stimulus file for 8-bit encryption with 1,500 traces

For simulations, the tools run a testbench Verilog file in Vivado several times with the plaintexts and key described in the stimulus file. A Value Change Dump (VCD), which contains state changes for each wire at each time slice in the simulation, is retrieved from Vivado. For each trace, these state changes are converted into Hamming Distance, a common power leakage model which counts the number of state changes for each wire. Hamming Distance summed across every wire at some sampling frequency to produce the final set of traces.

For hardware power traces, an FPGA is simulated by the same file. Voltage is measured starting at the time of a trigger signal until the end of each encryption, for example. This power signal is amplified and filtered by analog circuitry. This process will be described in greater detail in a later section. A capture device records these measurements and stores them in a Hierarchical Data Format HDF5. This data set contains plaintext input, keys, and measured data. It is later read and converted into trace objects.

Now that the traces are available to MITRE’s side-channel analysis tools, some pre-processing and statistical analysis can occur in a series of steps. These steps will be described in the proceeding sections. But, in general, the first step will either estimate statistical moments for correlation and Welch’s t-test, or compute histograms for rank correlation and MI. The diagram below outlines the entire process from trace collection to analysis.

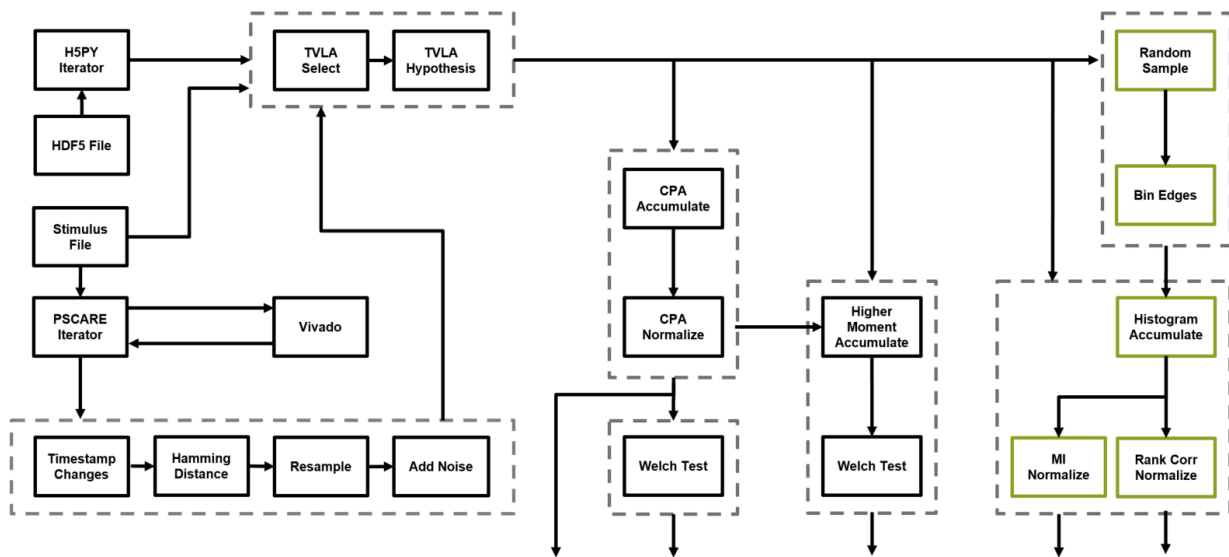


Figure 3.2: Overall system block diagram showing trace collection and final analysis. Dashed boxes represent Orchestrator objects, which pass each trace in a trace iterator through a series of steps. Output from the system comes from the CPA (correlation power analysis) Normalize step, Welch Test step, MI Normalize step, or Rank Correlation Normalize step. The original data source is either a PSCARE (Power Side Channel Analysis and Risk Evaluator) Iterator, which creates simulated HD traces, or an H5Py Iterator, which holds data from a power trace capture. The TVLA Hypothesis step yields the final set of traces for analysis. Newly implemented steps for this project are highlighted in green.

3.2.2 Uniform Random Sampling

Probability distribution functions are required for the total set of traces at every time sample, the two trace groups at each time, and the selectors for each trace (fixed vs random). PDF's can be estimated by histograms, which are composed of several data bins, each containing a count of data points which fall into that bin. Bin width estimation is a well understood problem in statistics. However, these estimation techniques typically requires access to the total population, which, in our case, is the n th sample from every trace. There could be thousands or millions of traces in the set, each containing hundreds or thousands of samples. So, it is not feasible to have access to the total set. A simple solution is to approximate the needed statistics of the entire set based on a small sample. To get a good estimate of the entire set, the small subset should be a uniform random sample, meaning that each trace has an equal likelihood of ending up the sample. The diagram below shows an algorithm for yielding a uniform random sample of traces of size M from the original set of size N . The Python NumPy random module was used to produce random numbers. It is seeded with a constant value before this step runs to ensure the same sample will be obtained on each run.

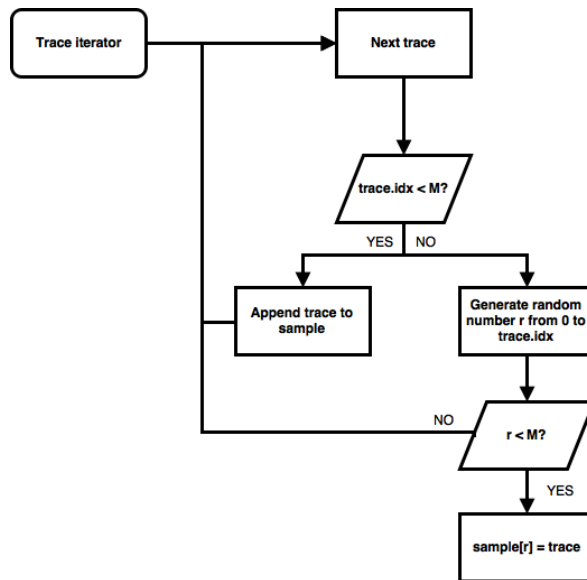


Figure 3.3: Diagram showing uniform random sample generation. A trace at index i is added to the sample at random index r if r , a random integer from 0 to sample size M , is less than M .

The probability that an element at index I will be inserted into the sample is:

$$P(Insert) = \frac{M}{I} \quad (3.1)$$

The probability that an element at index I is removed from the sample is equal to the probability that it is inserted in the first place, times the probability that any subsequent elements will evict this element. The probability that any subsequent elements will evict this element was determined through testing.

$$P(Remove) = \frac{M}{I} * \left(\frac{N-1}{N}\right) \quad (3.2)$$

The probability that an element at index I ends up in the final sample is equal the difference of the probabilities of insertion and removal.

$$P(Set) = P(Insert) - P(Remove) \quad (3.3)$$

Substituting the probabilities determined above, we end up with the following equation:

$$P(Set) = \frac{M}{I} - \frac{M}{I} * \left(\frac{N-1}{N}\right) = \frac{MN - (MN - MI)}{NI} = \frac{M}{N} \quad (3.4)$$

Since each element has an equal likelihood of being in the sample, it is a normal random sample. Of course, this assertion was tested and found to be true.

3.2.3 Histogram Construction

Now that the tools have a small sample to work with, it can go through the process of estimating optimal bin size and constructing histograms. The method of optimal bin width estimation used is the Freedman-Diaconis rule, which minimizes the difference between the area under the empirical PDF and the estimated histogram. Referring back to equation 2.14, it is shown that this rule requires interquartile range of each data set (nth time sample of every trace in the sample). This is difference between the 75th percentile and the 25th percentile. The numpy module in Python has a percentile function to calculate this value from the random sample. The population size used in the calculation is the total set of traces, not the size of the random sample. This process is performed on every time sample to produce a set of bin edges.

These bin edges are given as input to the histogram construction step. This step iterates through each trace and quantizes the time samples according to the existing bin edges. The quantized trace data is added to the data histograms and conditional histograms, and the selectors are added to selector histogram.

3.2.4 Mutual Information

The mutual information step simply computes entropy for the data histogram, selector histogram, and conditional histogram at every time sample to perform its calculation. Referring to equation 2.10, MI is calculated by the summing the entropy of the selector histogram and the n th sample's data histogram and subtracting the entropy of the n th sample's conditional histogram. Shannon entropy is the summation the product of each bin's probability and the logarithm of the probability.

To compute entropy, histograms are normalized by dividing by the sum of the histogram. Then, bins with zero probability are thrown out to avoid invalid results. The computation is then performed. The three entropy values required are used to compute mutual information at each time sample. This is the final MI vector.

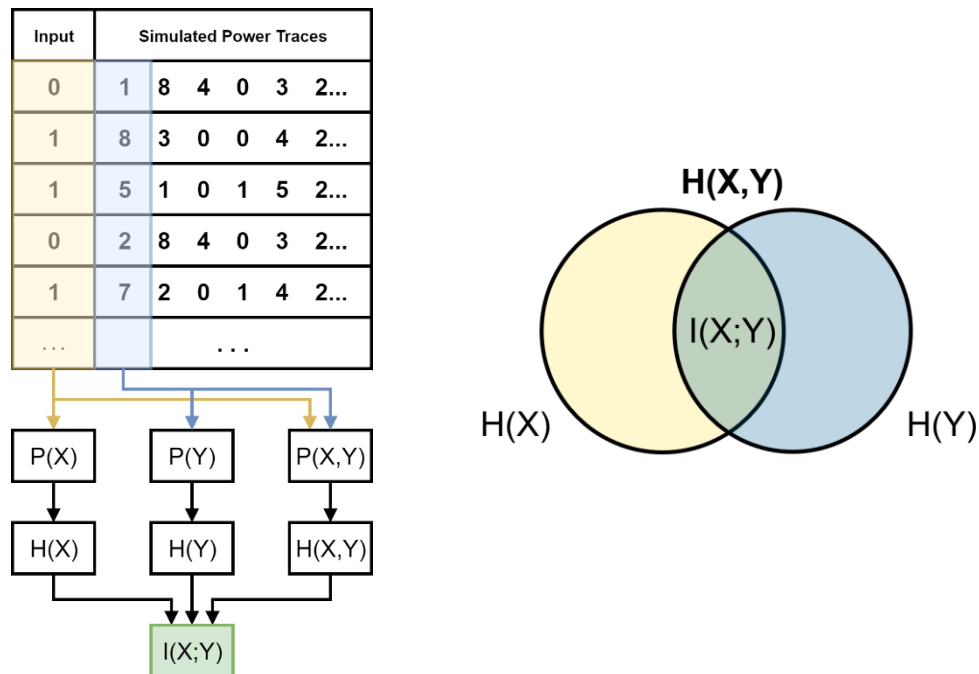


Figure 3.4: General algorithm for calculating mutual information $I(X;Y)$ from power traces and selectors. A selector histogram, data histogram, and

conditional histogram is created at for each time sample, highlighted in blue. Entropy is calculated and finally the MI equation is computed.

3.2.5 Rank Correlation

The rank correlation implementation uses the same histogram construction steps to produce categorical data. At each time sample, conditional histogram matrix is iterated through in order to count concordant and discordant pairs. The diagram below shows how quantized trace data at a single sample is transformed into a histogram and processed to count these pairs.

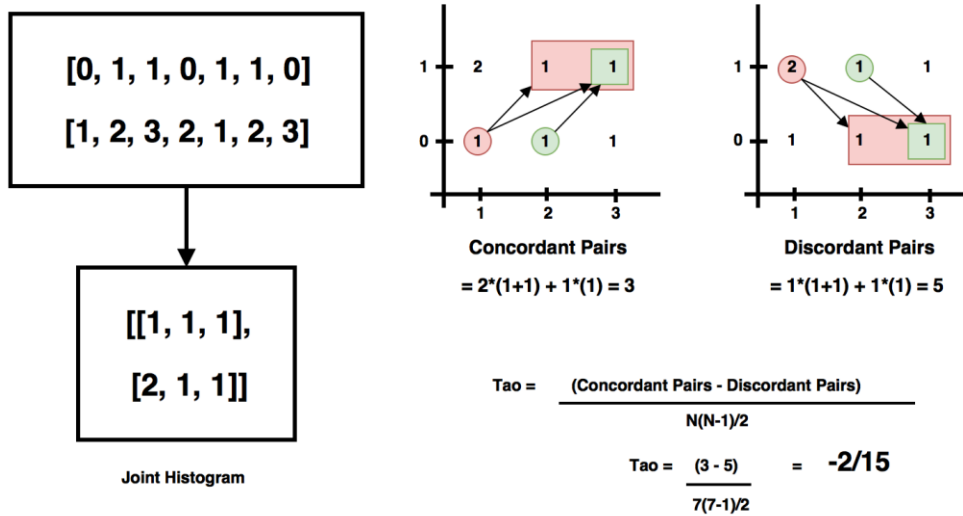


Figure 3.5: Categorical rank correlation process. On the left shows an example data set divided by selector value. This is used to create the conditional histogram below it. On the right shows an intuitive explanation of how the histogram matrix is processed to count concordant and discordant pairs.

At each element in the matrix, concordant pairs can be found by locating other elements which have a larger selector and larger data value or a smaller selector and smaller data value (positive slope on the graph). Discordant pairs can be found by locating other elements which are negatively sloped on a graph with respect to the current element. Ties are not considered.

3.2.6 Threshold Detection

It is important to be able to compare results of each statistical analysis. A standard method of signal significance detection helps compare each statistic vector and identify sources of leakage. Since

the TVLA methodology assigns a constant threshold of +/- 4.5 to the t-test result, this analysis is excluded from the threshold detection algorithm.

Permutation testing is a way of randomizing data by resampling without the assumption of a specific distribution. For the fixed vs. random test, a permutation test would involve assigning random selectors to each trace in addition to the actual selectors. The statistic vector using this second set of selectors should represent no dependency for the type of data collected.

The threshold value is based on the randomized statistic vector. From testing, it has been determined that some multiple of a high percentile of the randomized statistic vector appropriately represents statistical significance. For example, for an entire time vector, double the 95th percentile seems to be appropriate.

3.2.6 Hardware Power Trace Collection

Power trace collection must be carefully designed and carried out to obtain valid data. The diagram below shows a block diagram of the equipment setup for the procedure:

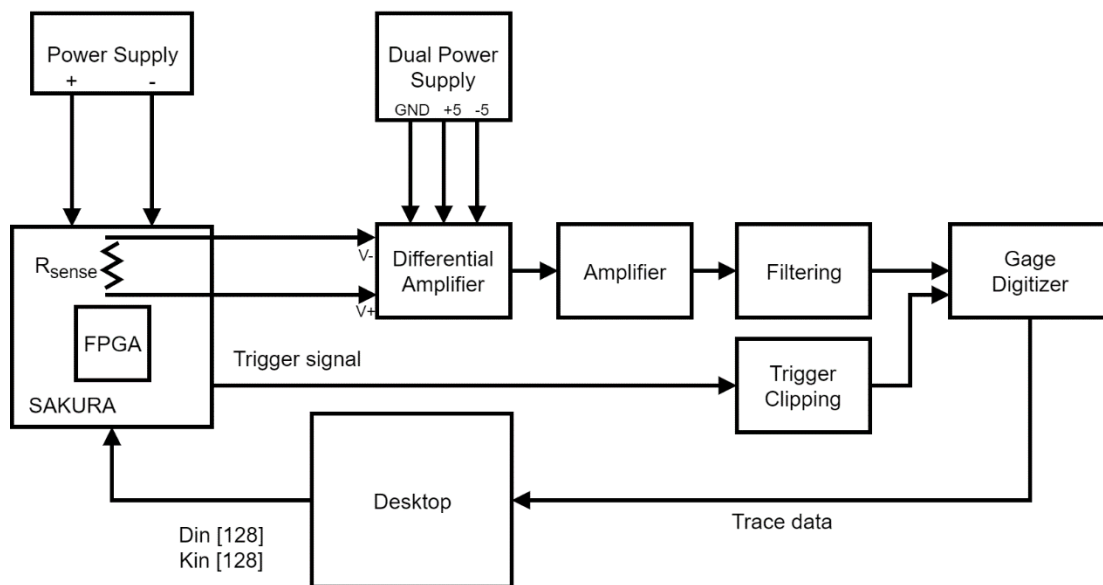


Figure 3.6: Lab setup for collecting power traces on the SAKURA GIII board.

From SAKURA-GIII evaluation board comes equip with a Kinex-7 FPGA and a voltage sense resistor. Voltage is probed at the two coaxial terminals of the sense resistor. A differential amplifier

board with the AD8129 chip amplifies this voltage signal. The differential amplifier is powered by two Acopian 5V DC power supplies in bipolar configuration. A second gain stage is inserted to bring the peak-peak of the signal closer to the ± 250 mV of the digitizer. This signal is passed through an anti-aliasing filter and wired to the Gage CS121G11U 12-bit USB digitizer. The filter is a Mini-Circuits 80 MHz low-pass filter, which is high enough to preserve most information in the signal and less than the Nyquist frequency of the digitizer. The trigger signal coming from the SAKURA board is processed by a trigger leveling board, which uses transistors to clip the signal at a lower amplitude to meet the GaGe's specifications. It is important to use shielded coaxial cables to reduce added noise and other EM interference. They should be as short as possible as well; Long cables produce an unwanted impulse response which widens the appearance of information leakage in time. Therefore, an orderly setup produces best results. Below is the final setup used.

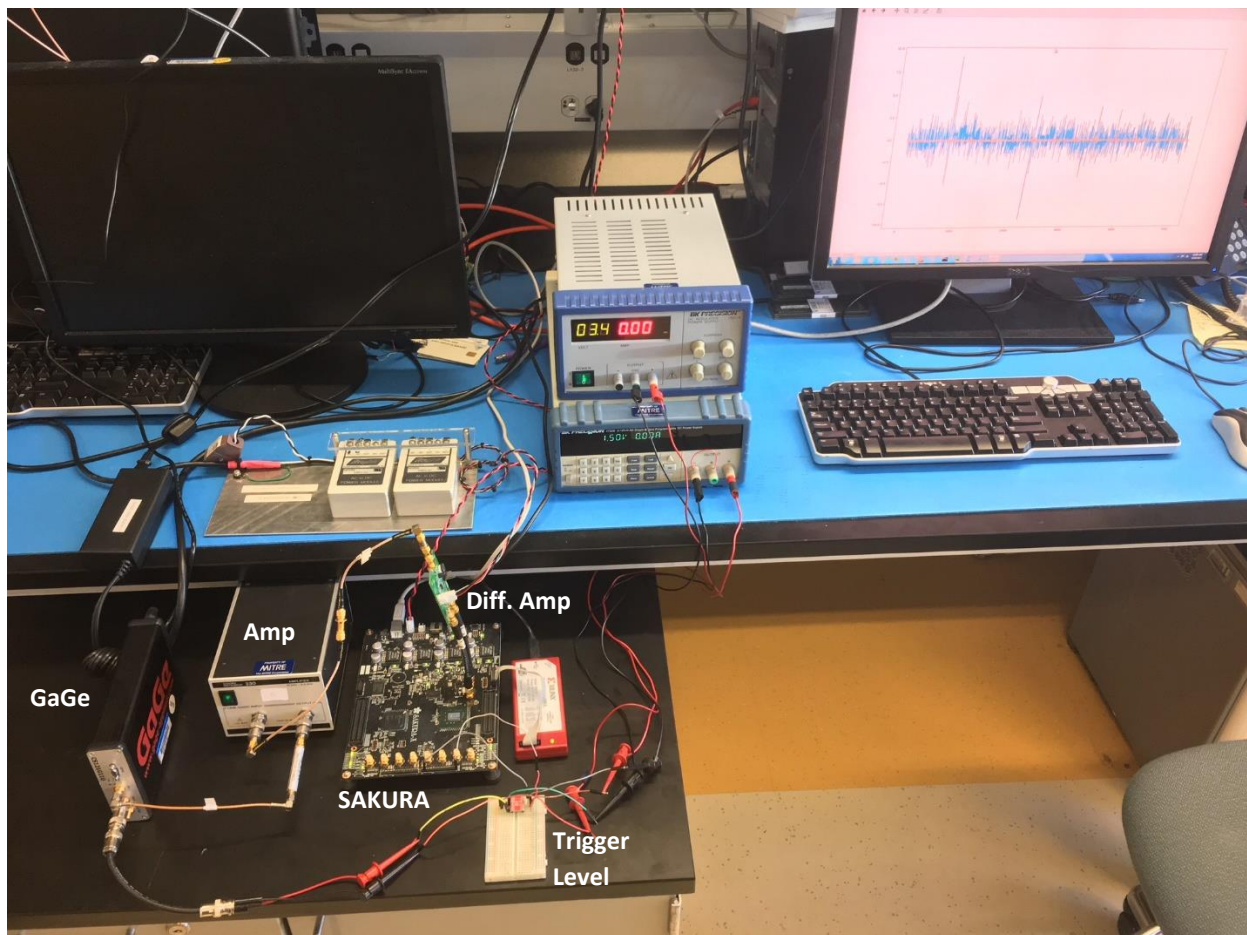


Figure 3.7: Lab setup for power side-channel trace collection. A desktop computer is running a python script which communicates with the SAKURA board and the GaGe digitizer.

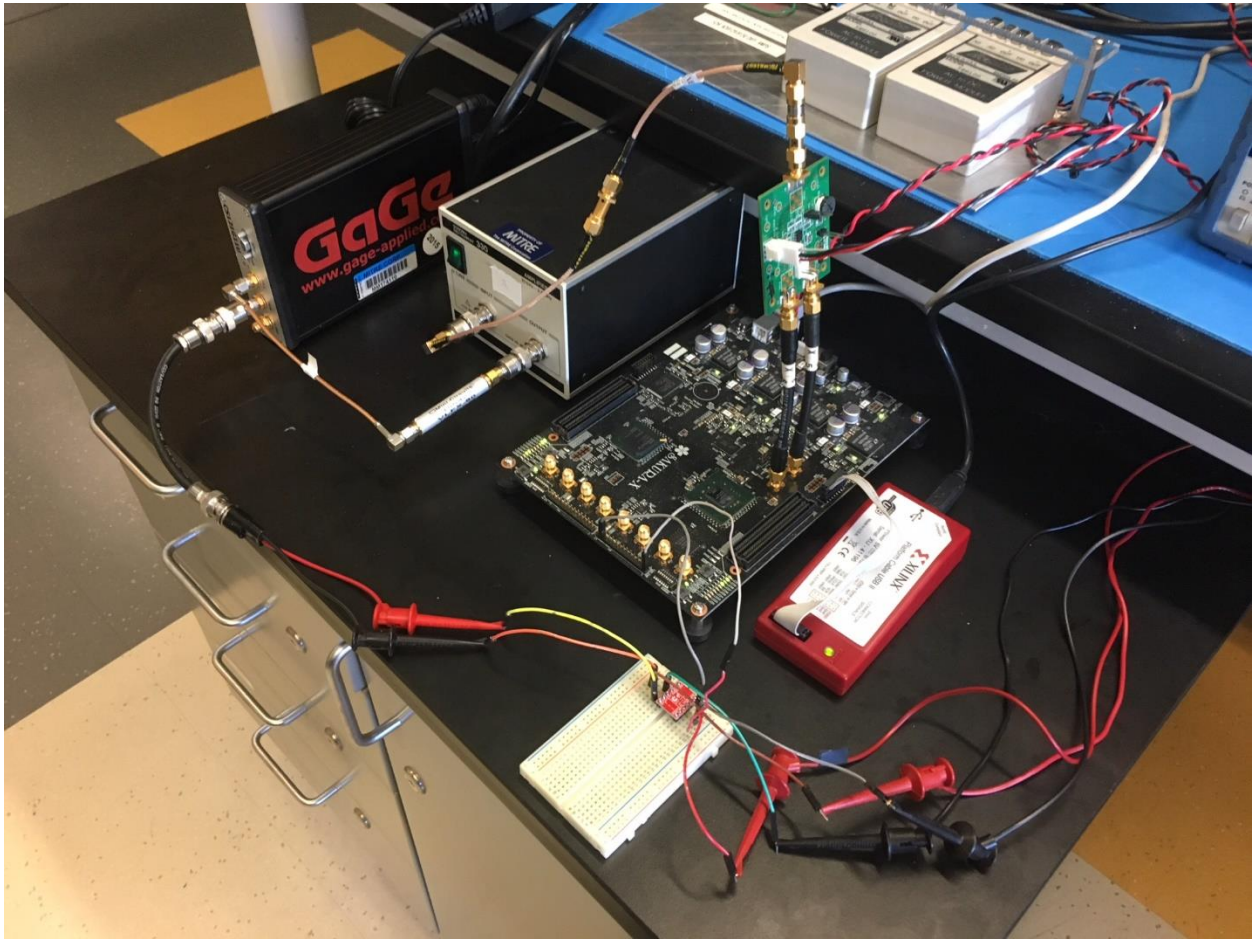


Figure 3.8: A closer view at the instrumentation used in the lab. The red XILINX platform connector receives data, key, and execution signals from the python program. It stimulates the hardware design on the FPGA, causing AES rounds to begin. Two coaxial cables connect the SAKURA sense resistor to the differential amplifier. The output of this amplifier is fed into a 2nd amplification stage. The output of this stage goes through an anti-aliasing filter before being digitized by the GaGe and transmitted to the computer.

3.3 Testing and Verification

3.3.1 Unit Testing

Extensive unit testing was performed on the statistical analyses to ensure the accuracy and advantages of each. The first test examines the performance of each discriminator for leakage in distinct moments. A data set was produced that contains two distinct distributions when divided by a binary selector, as exemplified below.

Dataset: [6, 7, 6, 1, 6, 0, 2, 2, 4, 7, 0, 4, 7, 0, 5, 4, 4, 6, 7, 4]
 Selectors: [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]

 Group 1: [6, 6, 6, 2, 4, 0, 7, 5, 4, 7]
 Group 0: [7, 1, 0, 2, 7, 4, 0, 4, 6, 4]

The histograms of each dataset are shown below with group 0 and 1 plotted separately:

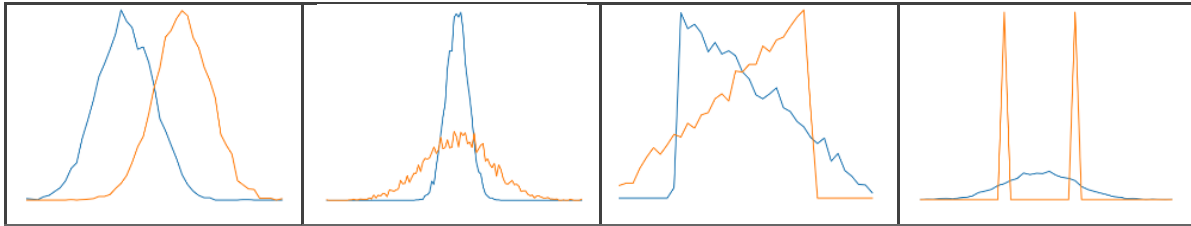


Figure 3.9: Histograms for the four datasets used in the moment varying unit test. The histograms shown here use 40 bins along the horizontal axis. The vertical scale is probability.

For each dataset, the two groups differ by some statistical moment. For example, the first dataset in Figure 3.9 contains two groups of samples that differ in average. The results of each discriminator were computed from 10 to 10,000 traces. The t-test should identify dependence for the dataset with a varying moment that corresponds to its target moment. For example, the 2nd moment t-test should identify dependence in the dataset with a varying standard deviation (the second moment) between the two groups. Linear correlation should only identify first order dependence, rank correlation should identify first and third order dependence, and mutual information should identify dependence in each dataset. This unit test also shows how population size influences each statistic vector.

The next test is almost identical, but the results of each discriminator for each dataset is computed for a range of noise amplitudes, not population size. The population was held constant at 2000 observations, while standard normal noise was added to the dataset from an amplitude of 0.1 to 100. This test is meant to demonstrate the effects of noise on each discriminator.

A test was devised to demonstrate that mutual information does not assume an attacker's strategy by estimating specific statistical moments. If there existed a device which leaks information through an unknown moment, or the moment is too high to estimate, the t-test is not as useful. MI however, will always be able to identify dependence in adequate conditions. To illustrate this fact, a dataset was devised that, when divided into two groups by a selector, yields two

different distributions with nearly identical moments. The function used to produce this dataset is the following lognormal function, where a is any real number:

$$f_a(x) = \frac{1}{x\sqrt{2\pi}} e^{-1/2(\ln x)^2} (1 + a \sin(2\pi \ln(x))) \quad (3.5)$$

This function will yield a different shape for each value of a , but will always result in the same infinite set of statistical moments. The figure below illustrates what the distributions look like:

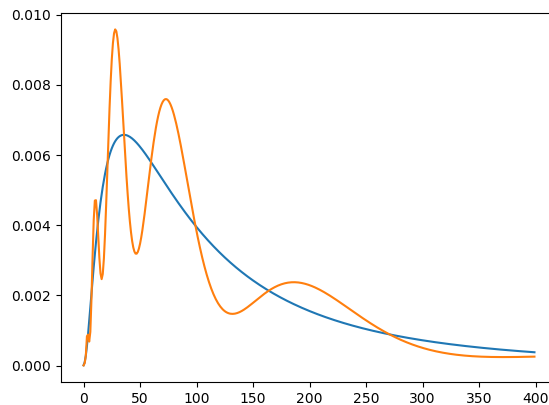


Figure 3.10: Two distributions with identical statistical moments but different shapes. These distributions were used to make a single dataset for analysis. The blue line has a value of 0 and the orange line has an a value of -0.5.

A set of power traces were fabricated with random noise at each time sample except for one sample, which has the distributions above, depending on the trace selector. So, at this one sample, there should be a dependency between the binary selector and trace data. Only mutual information is expected to identify this dependency.

3.4 Designs Under Test

3.4.1 AES Substitution Box

The first design used for testing was a substitution box, a nonlinear function that takes an N bit input plaintext and outputs a different N bit number. There is a one-to-one relationship between input and output. Substitution is the first step in an AES encryption round, so it can be used to model full AES by wiring the output result back into the substitution box module. This design takes an 8-bit plaintext, 8-bit key, and an input valid signal. These values are fed into an exclusive OR gate. At the start of the first round, stored in the intermediate register. Throughout each of the four rounds, the value in this register is directed into the substitution box and stored into the same register. At the end of the fourth round, the state of the intermediate register is stored in the output register. The diagram below shows a simple hardware design for this process.

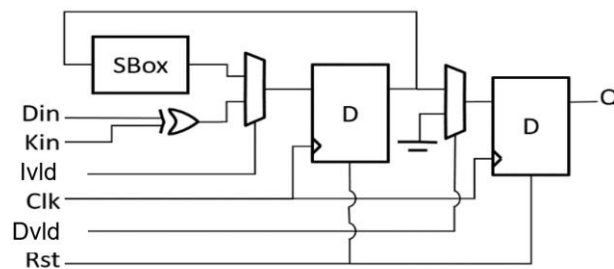


Figure 3.11: A simple logic diagram illustrating the s-box feedback design.

The *t-private* implementation will also be analyzed. As discussed earlier, each component in the design is copied and masks or masked component outputs are applied to the input of each copy. Figure 3.12 shows how the original design corresponds to the synthesized masked design for a single bit.

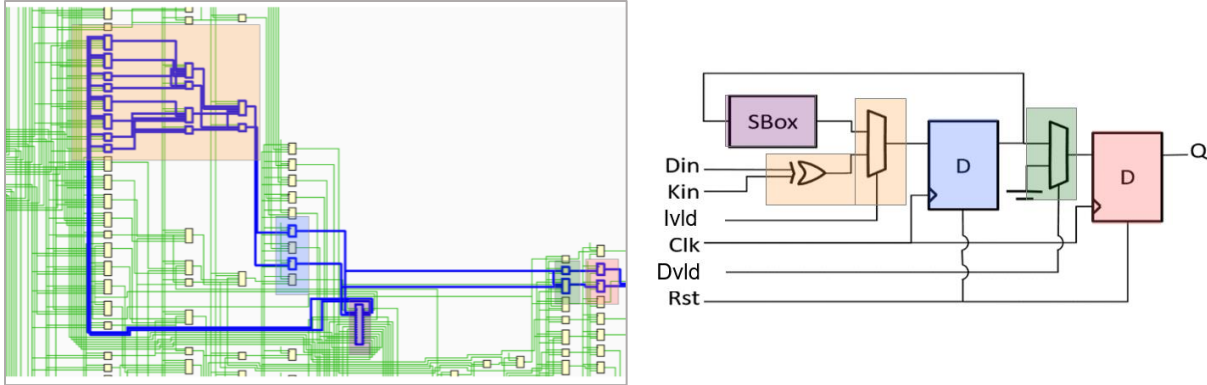


Figure 3.12: Post synthesis schematic for the *t*-private s-box feedback design with accompanying block diagram. Looking at the s-box state register (highlighted in blue) you can see that the *t*-private transform causes two duplicate registers: One holds the masked value of the s-box output and the other holds the mask bit of the s-box output.

The waveform below shows each wire and register in the testbench, along with other logic values, throughout the simulation. The total length of the simulation is about 2000 ps. The first stage of the simulation creates and applies masks. There is considerable glitching of multiple values in the simulation. These should produce leakage, as they involve input data. The next stage includes the data-key XOR and initial s-box state loading. Next are the actual rounds of the design. Last, the s-box state register is directed into the output register.

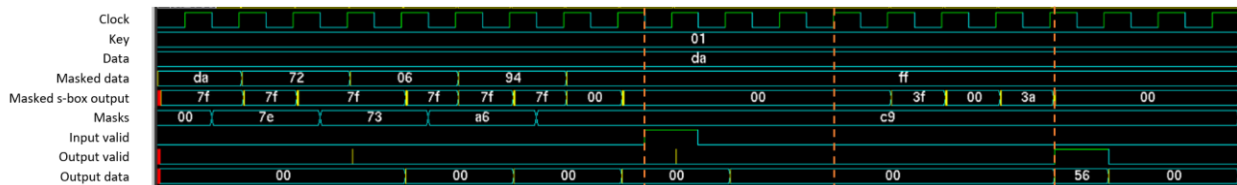


Figure 3.13: Testbench simulation for the s-box feedback loop design. Input plaintext is the 8-bit signal has the value $8'hDA$ in this simulation. The key is $8'h01$. The actual rounds occur in the first segment of this waveform. The masked data starts out as the original data value and undergoes masking to the value $8'hFF$ in the first segment of this waveform. The masked s-box output shows undesired glitching. We expect to see leakage at each of these glitches and during each round.

3.4.2 AES Encryption

AES-128 is the other design under test. AES takes a 128-bit key and 128-bit plaintext input. It performs 10 rounds total. Ten round keys are created from the original key. The initial round only adds the round key to the AES state using an XOR. The next 8 rounds perform nonlinear

substitutions, row shifting, column mixing and round key addition. The last round excludes column mixing.

In the design under test, there are at least 12 locations where leakage should be identified: data-key XOR, loading the initial AES state, and 10 rounds. Below is an example testbench simulation used for Hamming Distance calculations:

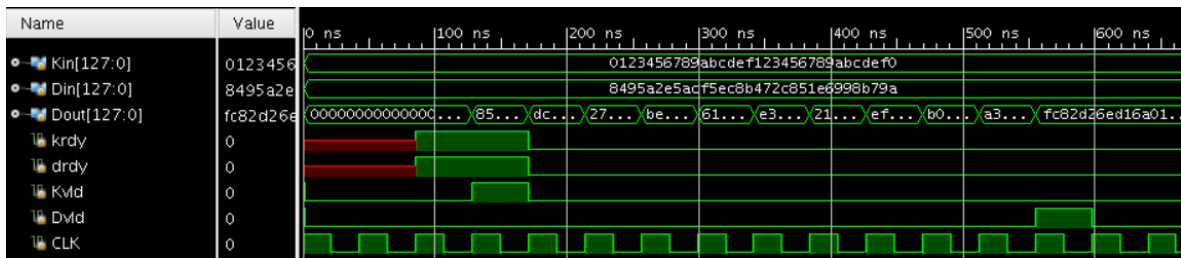


Figure 3.14: Testbench simulation for the AES design. The key (*Kin*) and data (*Din*) are constant throughout the simulation. The output data updates after each round. The encryption is initiated by the key-ready (*krdy*) and data-ready (*drdy*) signals. When AES finishes the last round, the data-valid signal is set high.

4 Results and Discussion

4.1 Unit Testing Results

The first unit test yielded results aligning with theoretical predictions. In this test, four datasets were created that, when divided into two groups by a binary selector, produce a pair of distributions that differ in some statistical moment. For each dataset, the suite of statistical analysis was performed. The expected results are outlined in the table below:

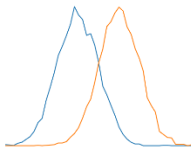
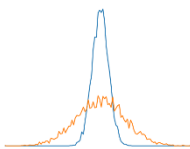
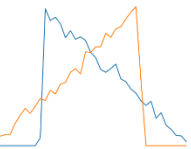
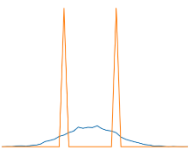
				
Welch's t-test (1)	Hit	Miss	Miss	Miss
Welch's t-test (2)	Miss	Hit	Miss	Miss
Welch's t-test (3)	Hit	Miss	Hit	Miss
Welch's t-test (4)	Miss	Hit	Miss	Hit
Pearson's coefficient	Hit	Miss	Miss	Miss
Rank correlation	Hit	Miss	Hit	Miss
Mutual information	Hit	Hit	Hit	Hit

Figure 4.1: Table illustrating expected results of the varying moment unit test

The actual results are shown in the figure below. Each expectation was backed by these results. Each t-test identifies dependence in datasets exhibiting its targeted moment variation. Third and 4th moment t-tests also identify dependence in the 1st and 2nd datasets respectively, which are inherently dependent in these higher moments.

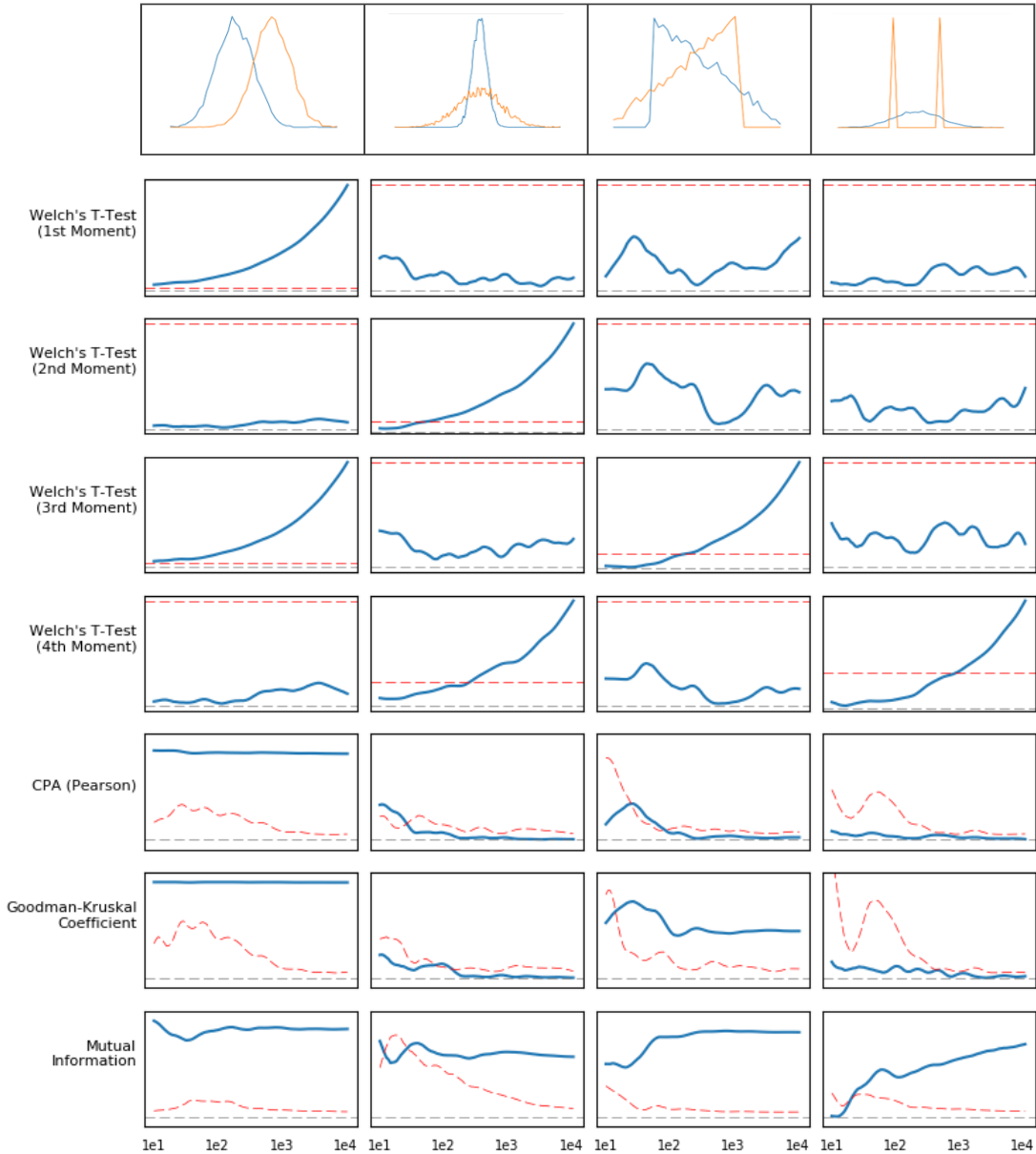


Figure 4.2: Experimental results of the varying moment unit test. Results are plotted over population size from 10 to 10,000 samples. The red dashed line indicates a threshold. For each t -test, the threshold is a constant 4.5. The threshold for the remaining tests are results of a selector permutation multiplied by 2.

This test proves the functionality of each statistical coefficient as predicted. Mutual information identifies dependence between data and selectors for each moment tested, whereas the t -test can only identify dependence within its target moment. Notice that the t -statistics will increase with

population size while correlations and mutual information settle to a constant offset. However, since the t-statistics' threshold remains constant while the others' decrease with population size, the statistical significance of each analysis result is population size dependent. This unit test also proves that rank correlation does in fact identify dependence when the two groups have the same mean but are not symmetric.

The next unit test was similar except that population size was kept constant at 2,000 samples and results were obtained for a series of noise levels. As expected, each analysis yields diminished results as SNR decreases.

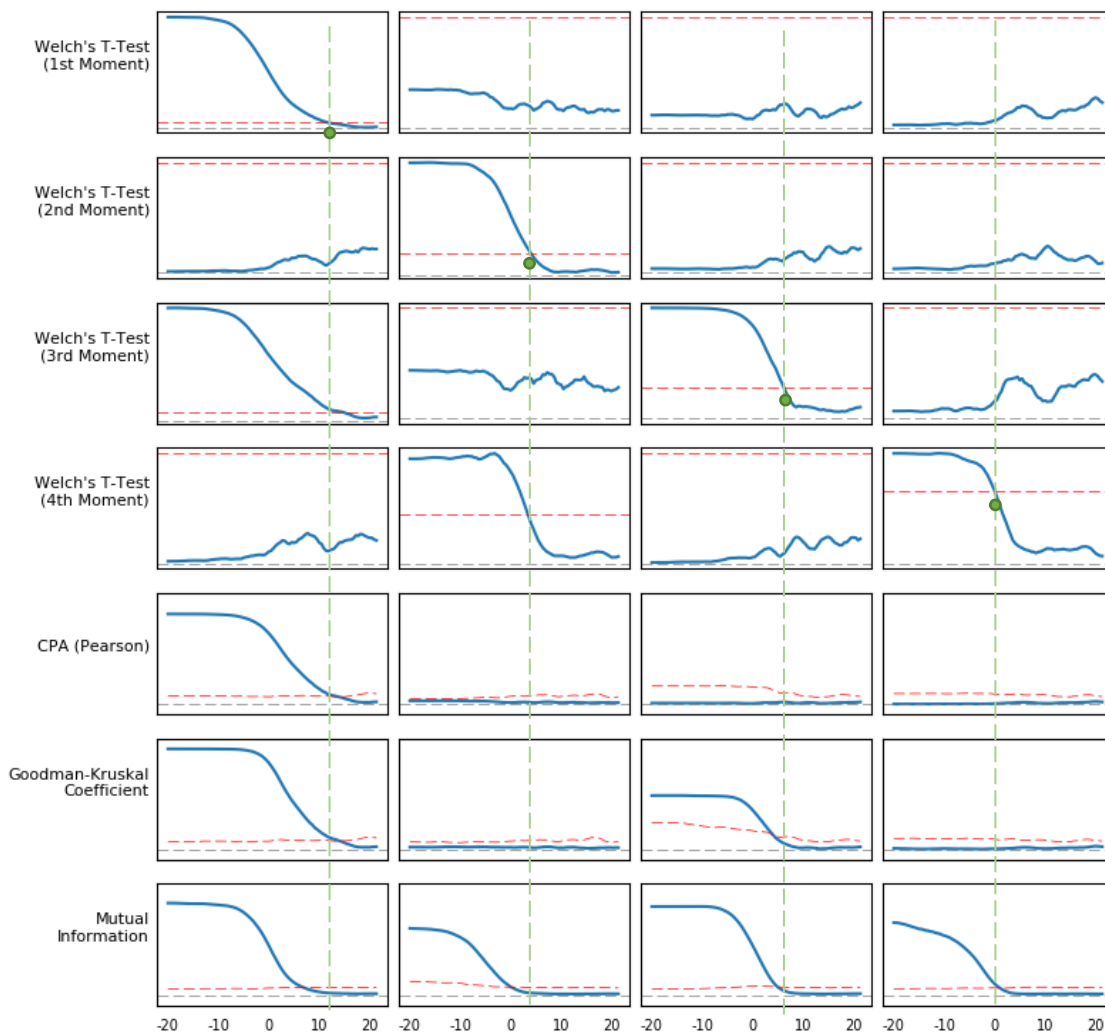


Figure 4.3: Experimental results of the varying moment unit test plotted over standard normal noise amplitude in decibels. The green dashed lines indicate where the respective t-test result becomes statistically insignificant.

This test does back up the theory that mutual information is most negatively impacted by noise, at least for 1st – 3rd moment leakage. This is observed when looking at the noise level where each analysis result dips below its respective threshold. Noise reduces the influence that the power signal has on entropy and therefore causes an underestimated MI result. This may be indicative that mutual information will not perform well for hardware power traces, which will suffer from higher noise than Hamming Distance simulation.

This next unit test illustrates a few key differences between the t-test, correlation, and mutual information. The figure below shows several line series on each plot; Each series represents the same statistic value from 0 to 200 samples and is associated with a different set of selectors. One set of selectors was used to create two distinct groups from a single dataset and the rest are random binary vectors. The line series derived from the actual selectors and data should distinguish itself from the other line series if the analysis type can identify the exhibited dependence.

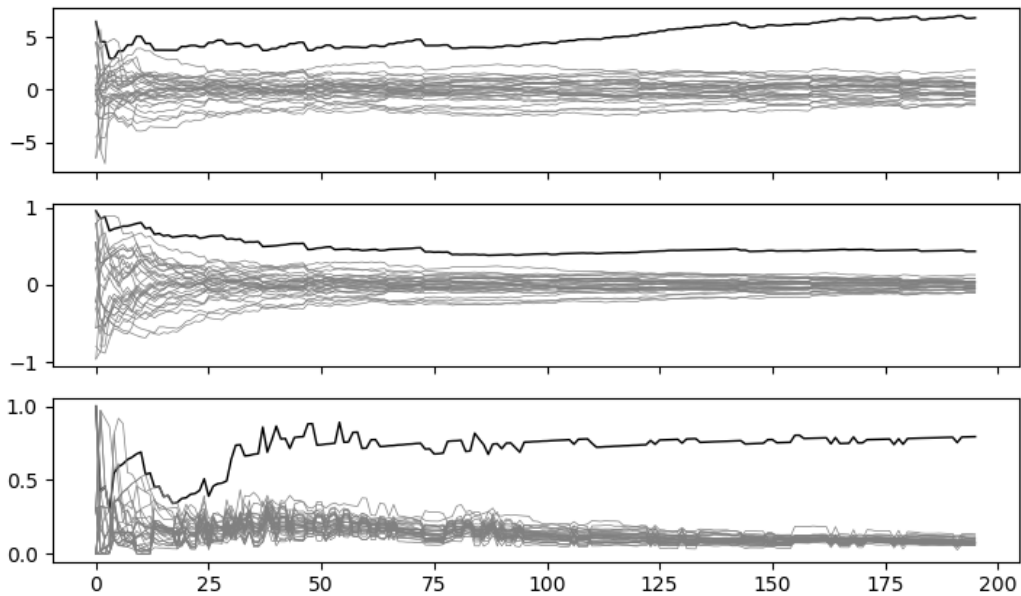


Figure 4.4: Plots for the t-test, Pearson's coefficient, and mutual information respectively over 200 traces for several sets of insignificant selectors (gray) and one significant selector (black). The dataset exhibits first and second moment leakage

To simulate a fixed vs. random test, a dataset was created that, when broken into two groups, contains a Gaussian distribution with a large variance and a distribution with very little variance.

The Gaussian distribution represents traces from the random group and the other represents traces from the fixed group, which shouldn't exhibit much variation under low-noise conditions. When the means of the two groups differ, the results from figure 4.3 occur. When the means are the same, the analyses yield the following results.

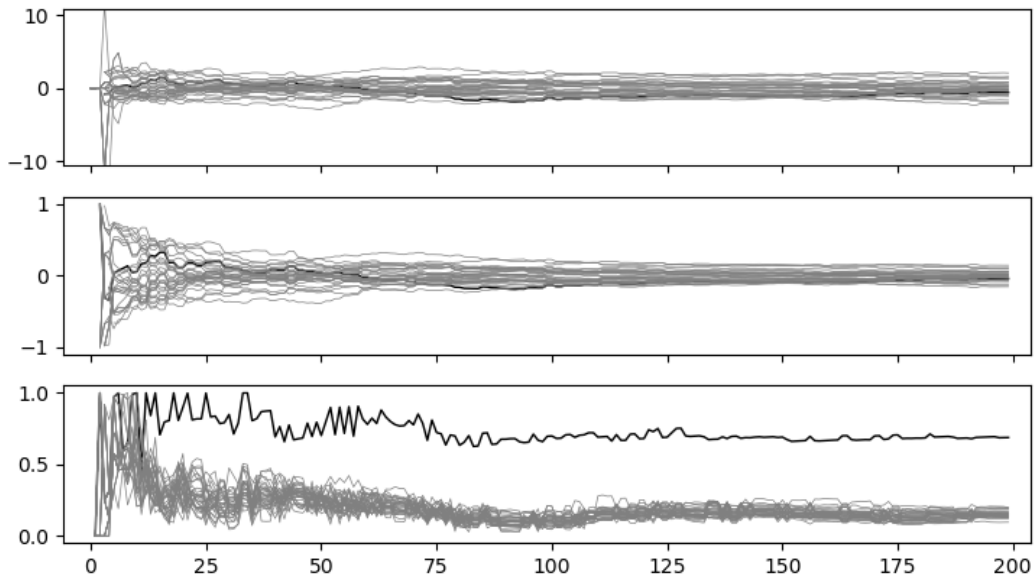


Figure 4.5: Plots for the t-test, Pearson's coefficient, and mutual information respectively over 200 traces for several sets of insignificant selectors (gray) and one significant selector (black). The dataset exhibits second moment leakage, but not first moment leakage.

The first figure, where the mean of the fixed group is different from that of the random group, shows that mutual information takes more traces to yield statistically significant results than the t-test and correlation. Significance is judged after the black line diverges from every gray line. For the t-test and correlation, this occurs and roughly 10 samples. For MI, it takes about double the number of samples. However, it is also observed that MI produces more quality results at higher population sizes; that is, the relative difference between significant and insignificant results is greater for mutual information. Further, the second figure shows that if the mean of the fixed group happens to be the same or close to that of the random group, the t-test (1st order) and correlation will not yield a significant result whereas mutual information will.

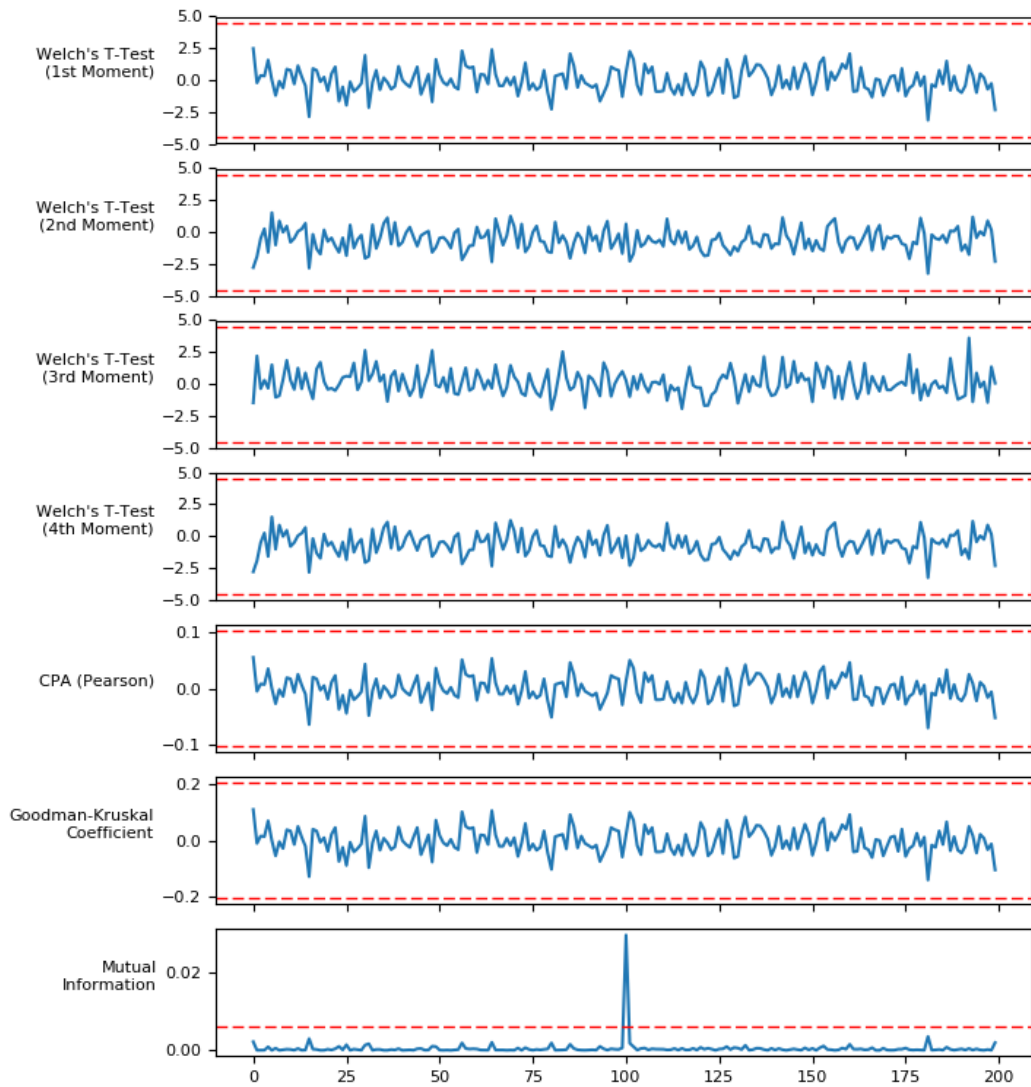


Figure 4.6: Statistical analysis results from 10,000 artificial observation traces where there is a single point of information leakage. Notice that only mutual information can identify this leakage as it is not derived from data groups with varying moments.

4.2 Experiment Results

4.2.1 Substitution Box

The first design examined was the unprotected substitution, a basic component of symmetric key encryption algorithms which performs nonlinear substitutions of input bytes. As discussed earlier, this design performs four rounds of substitution where s-box output is fed back as the input.

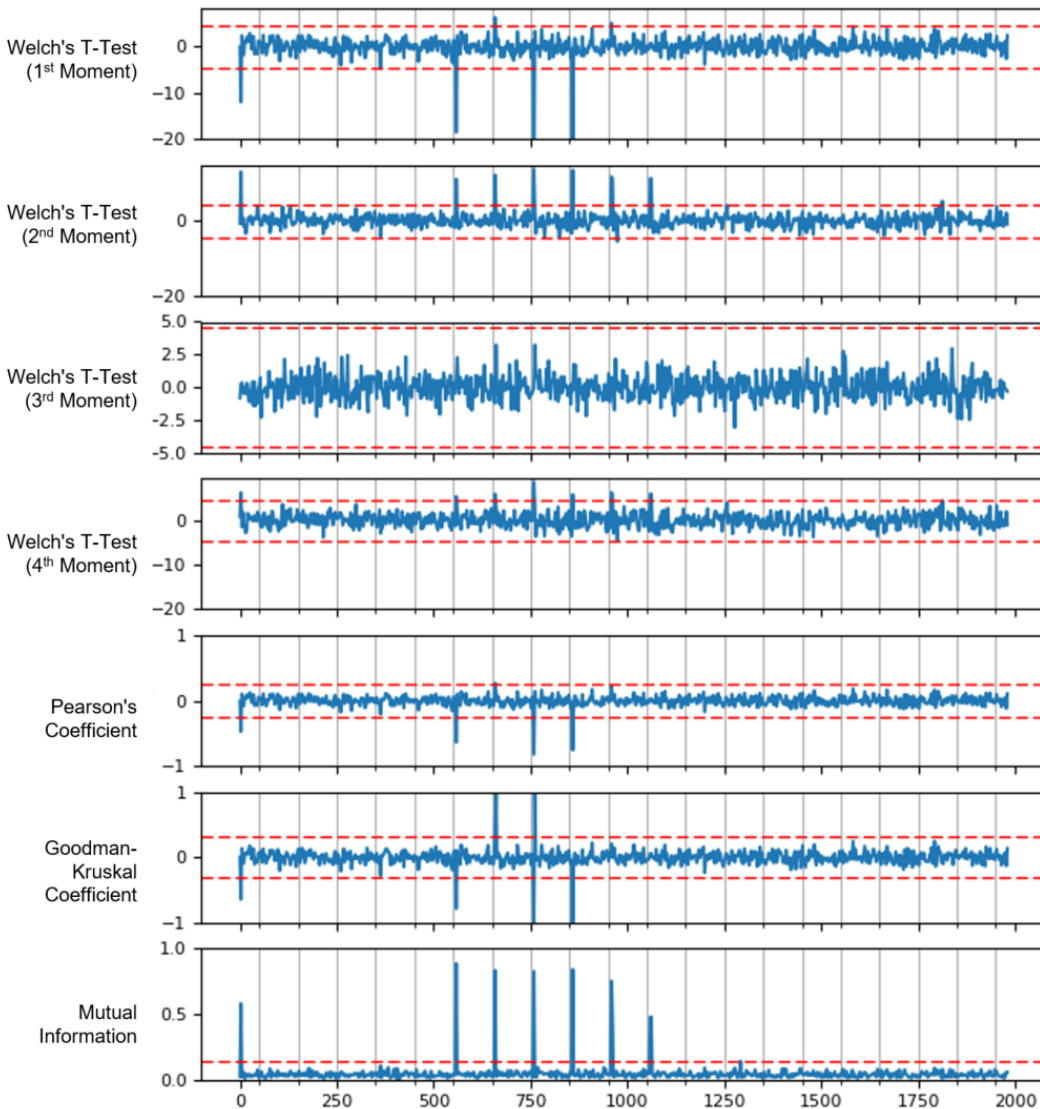


Figure 4.7: Statistical analysis results for an unprotected substitution-box feedback-loop design. The data is derived from a simulation about 2000 ns long.

Leakage “hits” should be expected at six consecutive rising clock edges: one for the initial s-box state write with data XOR key, one for each of the three rounds, one when the s-box output is written to the output register, and a final one when that output register is cleared. The number of wire state changes at each of these occurrences are influenced by input data. The results are shown above, where the timescale is in nanoseconds.

First, notice the difference between Pearson’s linear correlation coefficient and Goodman-Kruskal’s categorical rank coefficient; They are nearly identical at most times, but differ where there is identified leakage. The fact that they are so similar is an assurance that the histogram estimator produced adequate bin edges for the dataset. More importantly, rank correlation clearly provided stronger hits than linear correlation. This may be because rank correlation can identify third moment dependence, which is present at the same time locations of the affected hits.

Next, examine the mutual information vector; With 1,000 traces, this statistic produced the most distinct hits. Further, it identified leakage of any moment at every expected location. This is an advantage over the t-test, which requires a target moment.

A good example is the last leakage source at time 1050 ns where the fixed group has a narrow distribution, and the random group exhibits a wider Gaussian distribution. This is expected because the constant input should produce a constant Hamming Distance, and variable input should produce variable HD. In most cases, this leakage would be caught by both the 1st and 2nd moment t-tests. However, just by random chance, the fixed group impulse and the random group Gaussian distributions have the same mean. This means that only the 2nd moment t-test should identify this leakage because the standard deviations are very different. Thus, the validity of the t-test is skewed by the choice of fixed input during a fixed vs. random test. MI identifies this leakage because the entropies of the fixed and random groups are differ compared to that of the overall dataset. This illustrates the flexibility of mutual information.

4.2.2 t-private Masked Substitution-Box

The *t-private* masking countermeasure does not appear to protect this design. The post-implementation timing simulation yields HD vectors that are dependent on input data at multiple times in the simulation. This is accredited to unequal routing delays between masked components and unmasking caused by feedback loops. Every analysis identifies each leakage source up to 250

ns. These are caused by unmasked operations with input data. Mutual information identifies leakage at a three clock edges that are not shown in any other test before 1000 ns. And, as before, it shows strong leakage during the input multiplexing, s-box rounds, and output register writing. There appears to be some third moment leakage present as well. Also, the Goodman-Kruskal rank correlation coefficient shows strong correlation at clock edge not identified by Pearson's coefficient which also exhibits third moment leakage.

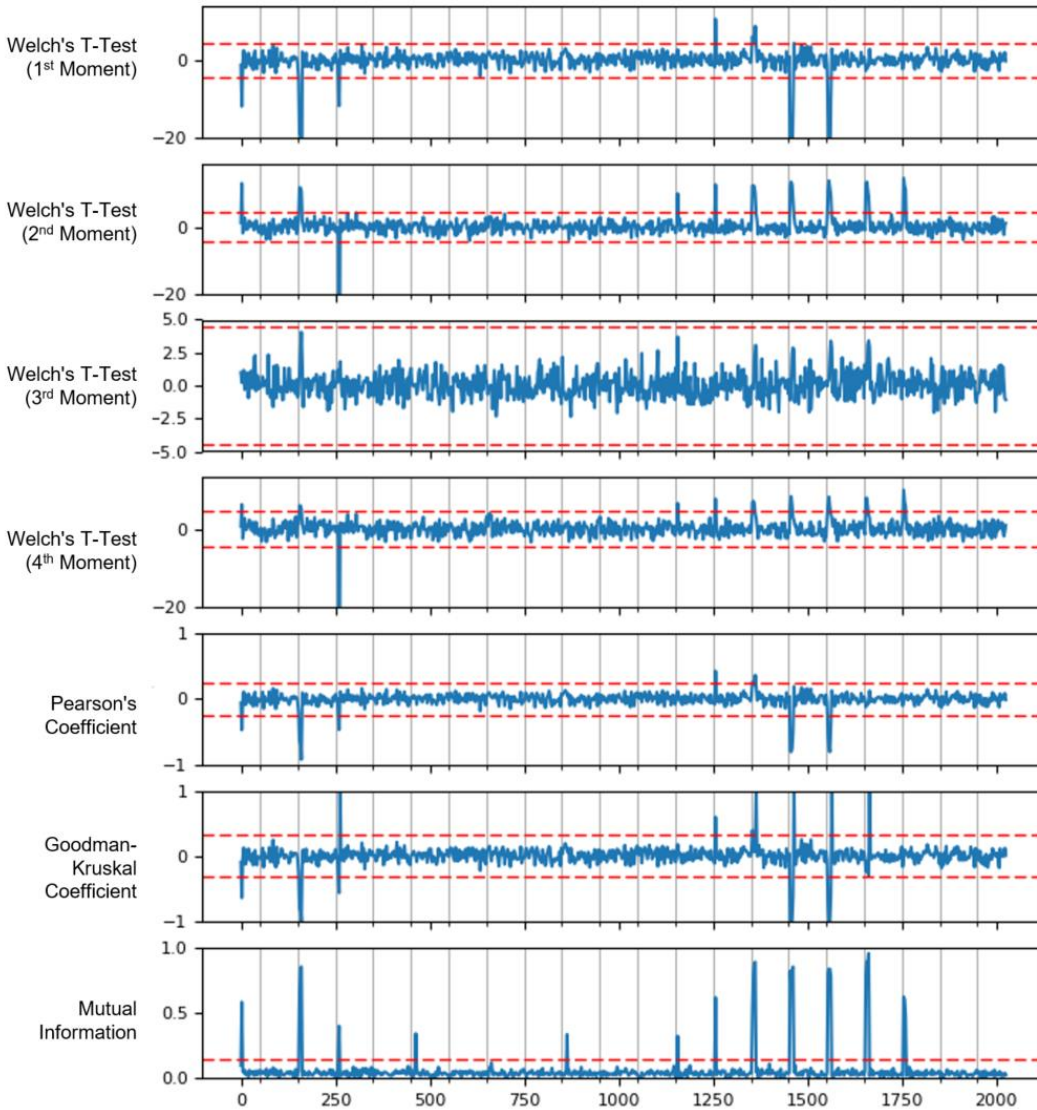


Figure 4.8: Leakage evaluation of a t -private masked substitution-box feedback design including Welch's t -test, correlation, and mutual information.

4.2.3 AES Encryption (Hamming Distance Simulation)

Twelve points of leakage are expected for the AES encryption simulation. The last 10 are the actual AES rounds. The first is simply a result of setting the value of the input plaintext data in the Verilog testbench file. The second is from the XOR of this input plaintext with the secret key. Looking at the following figure, this AES implementation is potentially insecure from side-channel attacks.

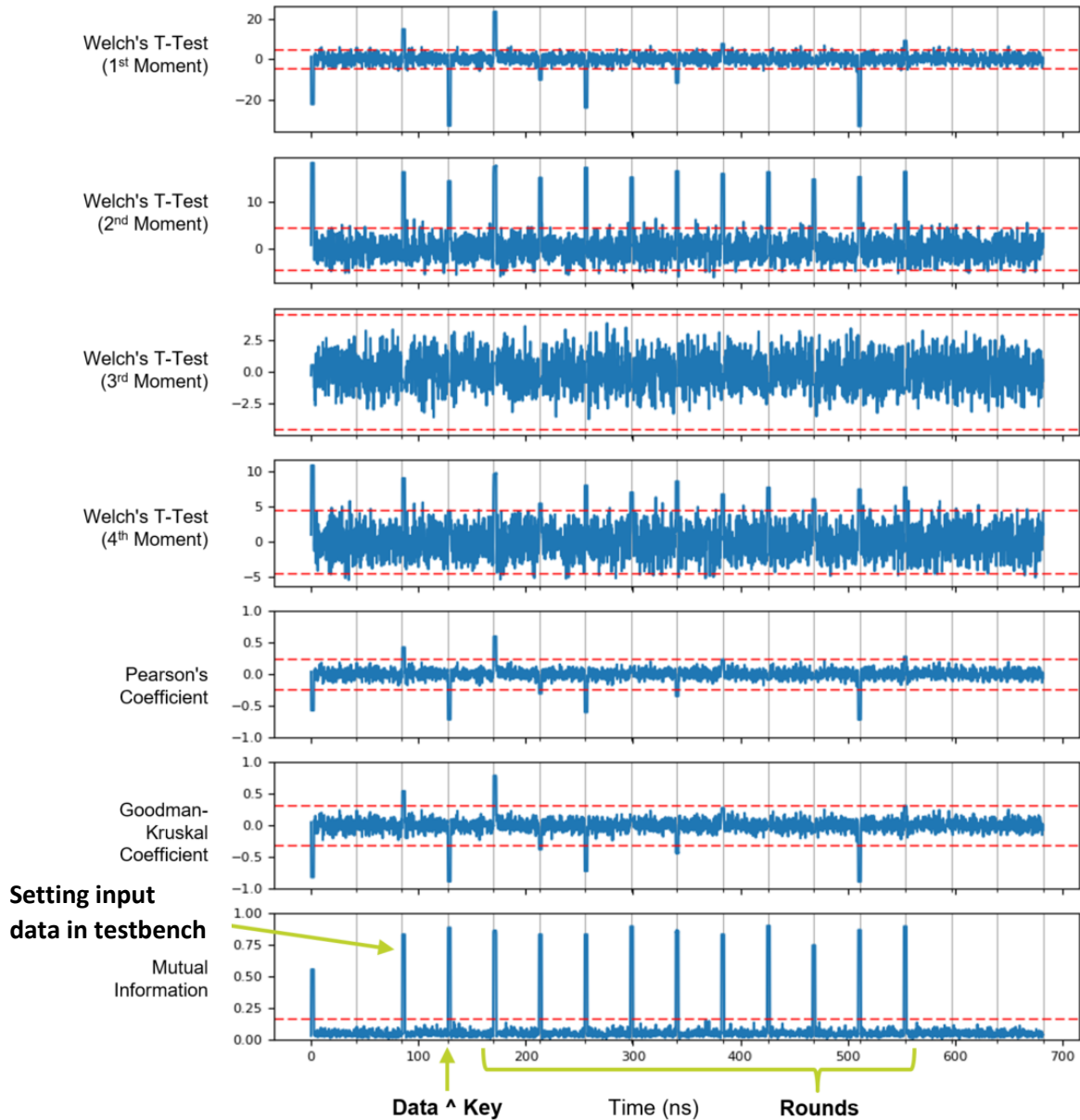


Figure 4.9: AES encryption leakage assessment performed by Welch's t-test, higher moment t-tests, correlations, and mutual information.

This design exhibits mainly 2nd moment leakage for the fixed vs. random test. Some first moment leakage is detected only when the fixed group's narrow distribution is centered far enough away from the random group's wider distribution. Since both distributions are roughly symmetrical, there is no 3rd moment leakage. Thus, the linear and rank correlation coefficient vectors are nearly identical. The mutual information vector reveals the most information leakage in this design. It discovers potential vulnerabilities at each expected clock edge regardless of the relative statistical moments between the two trace groups. It also produces the highest signal-to-noise ratio of any other discriminator.

A modified design examines a single round of AES. In this analysis, the sampling frequency is increased from 400 MHz to 4 GHz to highlight the subtleties of one round.

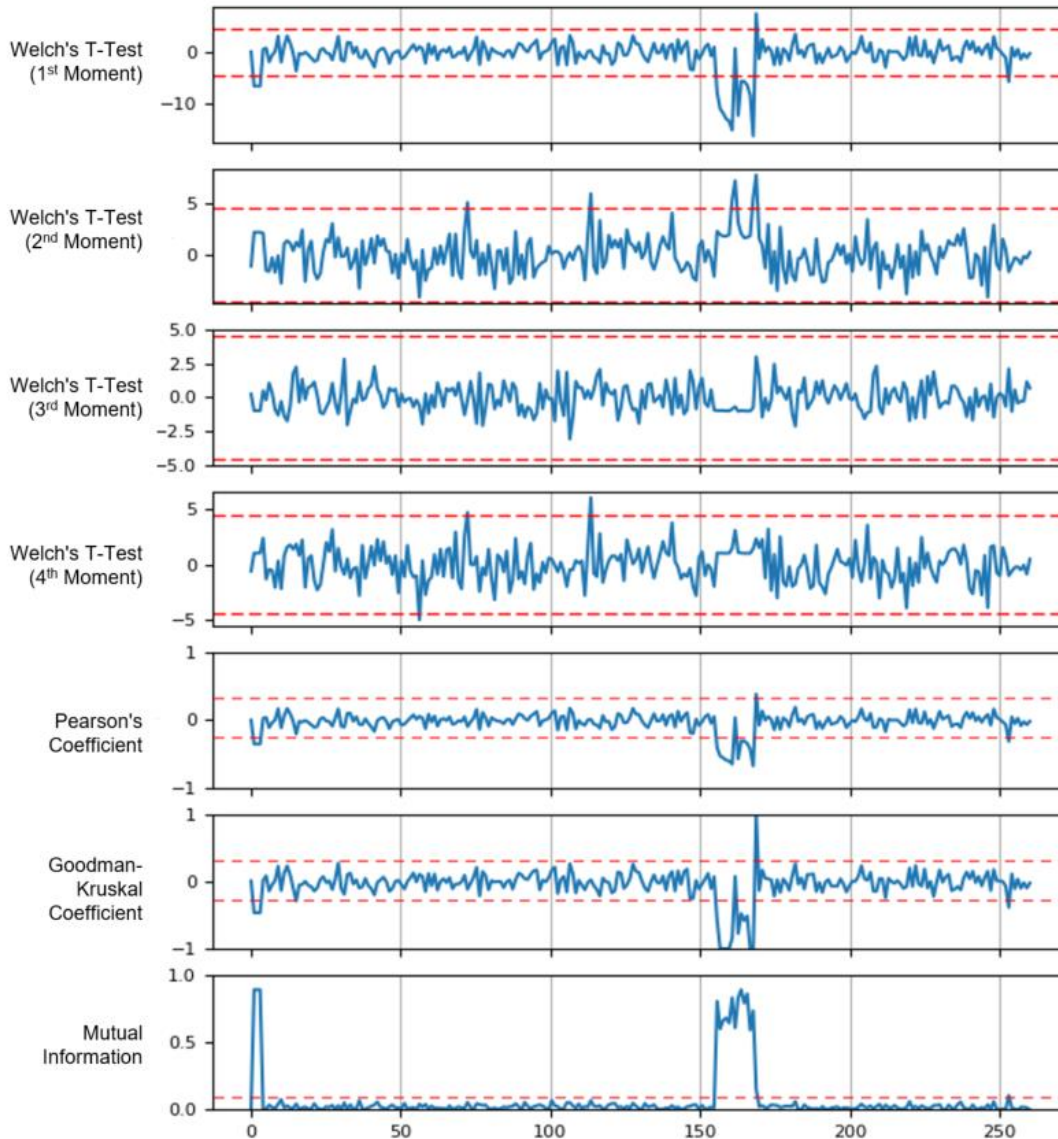


Figure 4.10: Leakage evaluation of a t -private masked substitution-box feedback design including Welch's t -test, correlation, and mutual information.

The round occurs at just after the 150-ns marker. Looking just at 1st and 2nd moment Welch's t -test, notice that both 1st and 2nd moment leakage is present. However, they occur at distinct times within the round. At the beginning of the round, only 1st moment leakage is revealed, followed by 2nd moment, 1st again, and then a combination of both. Mutual information, on the other hand, shows data dependence though the entire round much clearer than other tests. Another point of interest occurs around time 130ns; there is a "hit" on the 2nd and fourth moment t -tests. However,

we know this is a false positive because it does not occur on a clock edge and the simulated design shows no activity at that time. This most likely means that 1,000 traces are not adequate to provide an accurate t-test result. That population size is, however, enough to provide a strong MI result.

Up until this point we have only been examining the fixed vs. random test. However, it is common to evaluate this type of design with an algorithm-specific test. Tests that are specific to AES uncover leakage that is more likely to allow a key attack. For this test, the output of each AES round was calculated for each trace. The 2nd round is arbitrarily chosen as a target. Each trace's selector is derived from the exclusive-or between the 2nd round's input and output state. This 128-bit represents the Hamming Distance of the AES state register.

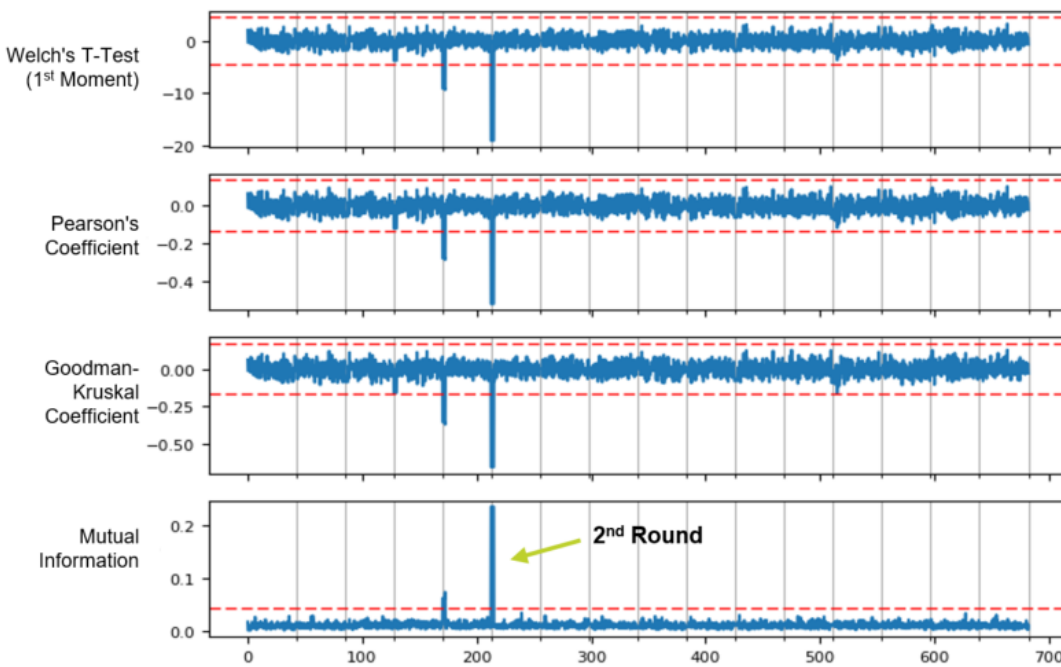


Figure 4.11: AES leakage evaluation results for an AES specific test targeting round 2 (time 215ns).

In practice, an attacker may only use one bit of this value as the trace selector. However, since we are performing leakage assessment it is more useful to use the entire state, even if this is not a practical attack. Below are the results of this test.

As expected, each analysis reveals leakage at the 2nd round shortly after the 200-ns mark. Since the selector uses output of the first round as well, we also see a smaller hit at the first round. This

should also continue back to the original state load, as seen in both correlations and the t-test. However, mutual information does not clearly identify very small differences in mean relative to the total spread of the two groups' distributions; thus, the first peak is lost with MI.

4.2.3 AES Encryption (Power Traces)

Theoretically, the results from this test should be comparable those of the simulated results. The leakage vectors under analysis are physical power measurements instead of Hamming Weight calculations from post-implementation timing simulations. The following figure shows what these traces look like.

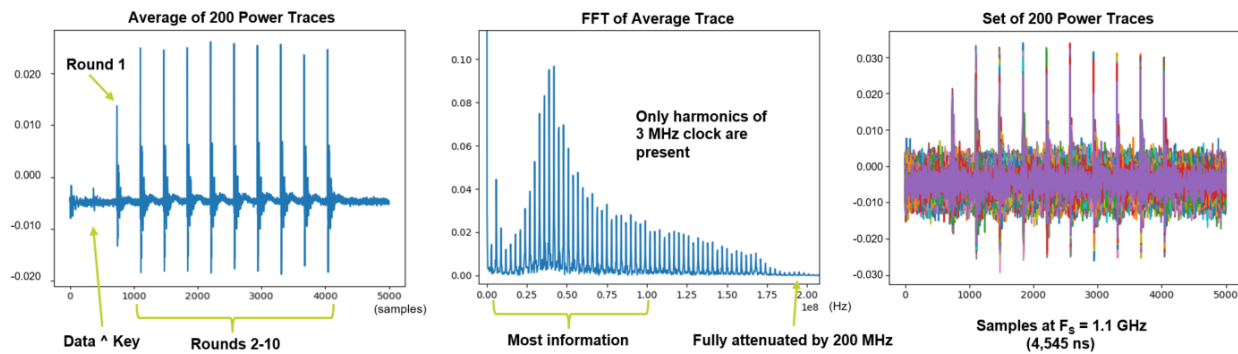


Figure 4.12: Plots showing average of 200 power traces (left), the frequency domain of this sample, and overlapping plots of this sample. At the clock edges of all 10 AES rounds, power consumption is much greater than variance from noise. With enough traces, this noise is eliminated on average. The first round consumes slightly less power as it only performs round key addition. The data-key XOR consumes a small amount of power, but not more than the amplitude of noise. The x-axis is number of samples at 1.1 GHz and the vertical axis is measured in Volts. Most information is found between 10 MHz and 100 MHz. Higher frequencies were filtered to avoid aliasing.

As shown in the figure, the trace contains about 11 clock cycles of interest. The first performs some setup including combining the data and key to produce round keys. The following 10 are the actual AES rounds. The first round, which occurs right before the 1000th sample, consumes visibly less power since it only performs round key addition. There is significant noise present, as the signal of interest is very small. The effects of this noise are diminished as trace count increases. The trace set must be large enough for the tools to confidently identify a difference in the two groups. With a set of 15,000 traces, a clear difference in mean is noticeable at the 5th round power impulse. The average of the fixed group, shown in blue peaks at a higher amplitude than the

random group. Looking at the histogram for the maximum voltage value of this round shows two distributions that clearly differ in mean.

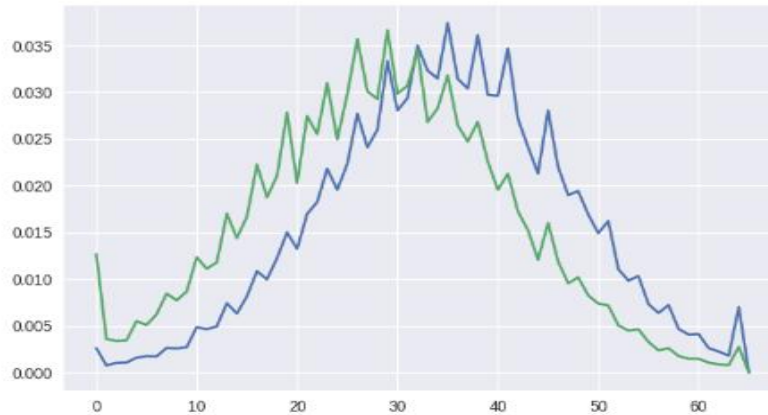


Figure 4.13: Plot showing that there at least exists first moment power leakage in this circuit. At the peak of round 5, this normalized histogram shows that the groups have different averages and roughly the same variance. Blue is the fixed group and green is the random group. The horizontal axis is bin number.

The results of two 5,000 trace sets (for a total of 10,000 traces) are shown below. They illustrate the difference seen in results quality after organizing the instrumentation setup.

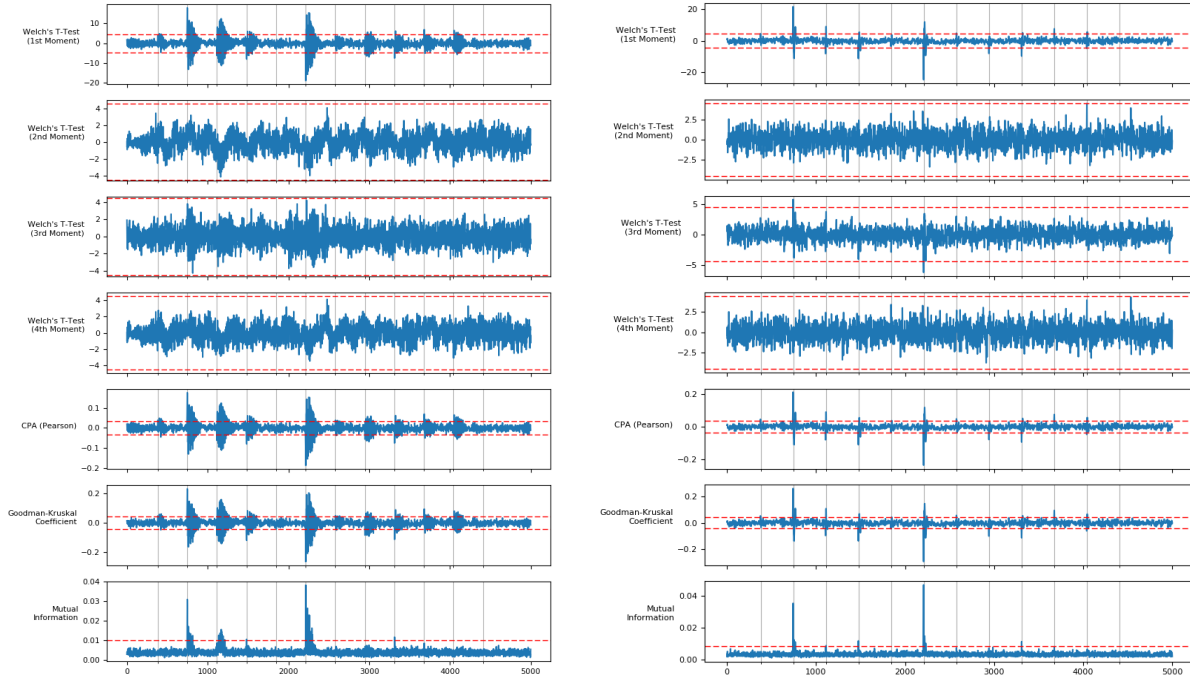


Figure 4.14: Full analysis results for the same AES implementation, both for 10,000 power traces. The instrumentation set-up remained largely the same, but was organized so shorter cables could be used without intersection or coiling. This yielded power traces that did not suffer as much from environmental noise or the coaxial cables' impulse response. The results from the new set of traces are stronger and more precise.

Each set of results show t-test hits at most of the expected clock edges. Mutual information however, is lacking in significant hits by comparison. As previously stated, mutual information may take more traces to yield significant hits. Therefore, another power trace capture was performed for a total size of 100,000 traces. The results of this collection for the fixed vs. random test is shown below:

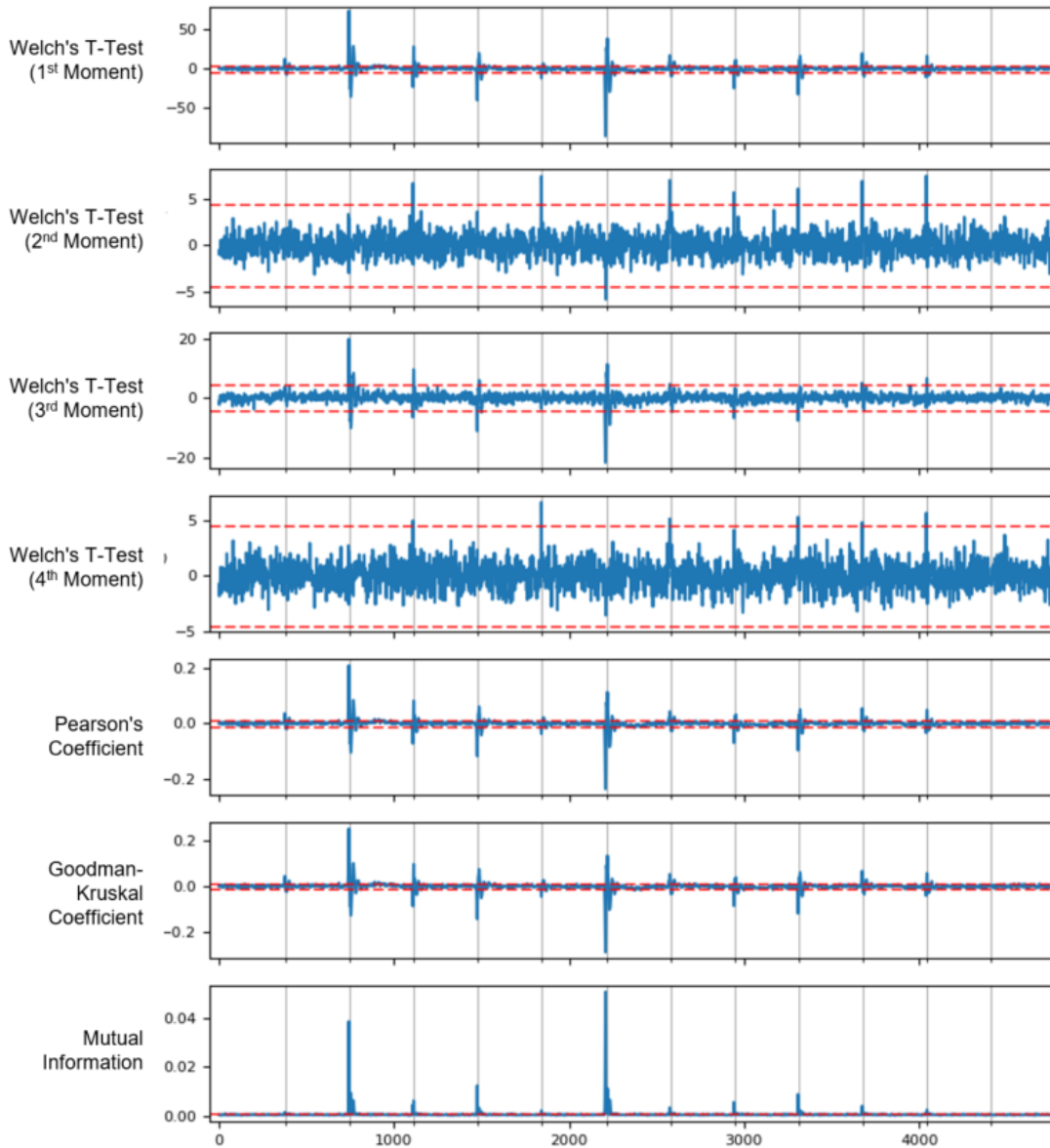


Figure 4.15: Results of power side-channel risk assessment with 100,000 traces. For this many traces, mutual information produces clearer hits than the moment targeting tests. The peak of each hit is very significant relative to its threshold.

With this many traces, we see that mutual information does reveal leakage at every expected location. In fact, the relative significance of each hit is much greater for MI than the t-test; The most significant hit is about 50 times greater than the MI threshold, compared to the same t-test

being only about 15 times greater. Scaling the y-axis helps see some of the smaller hits in each analysis.

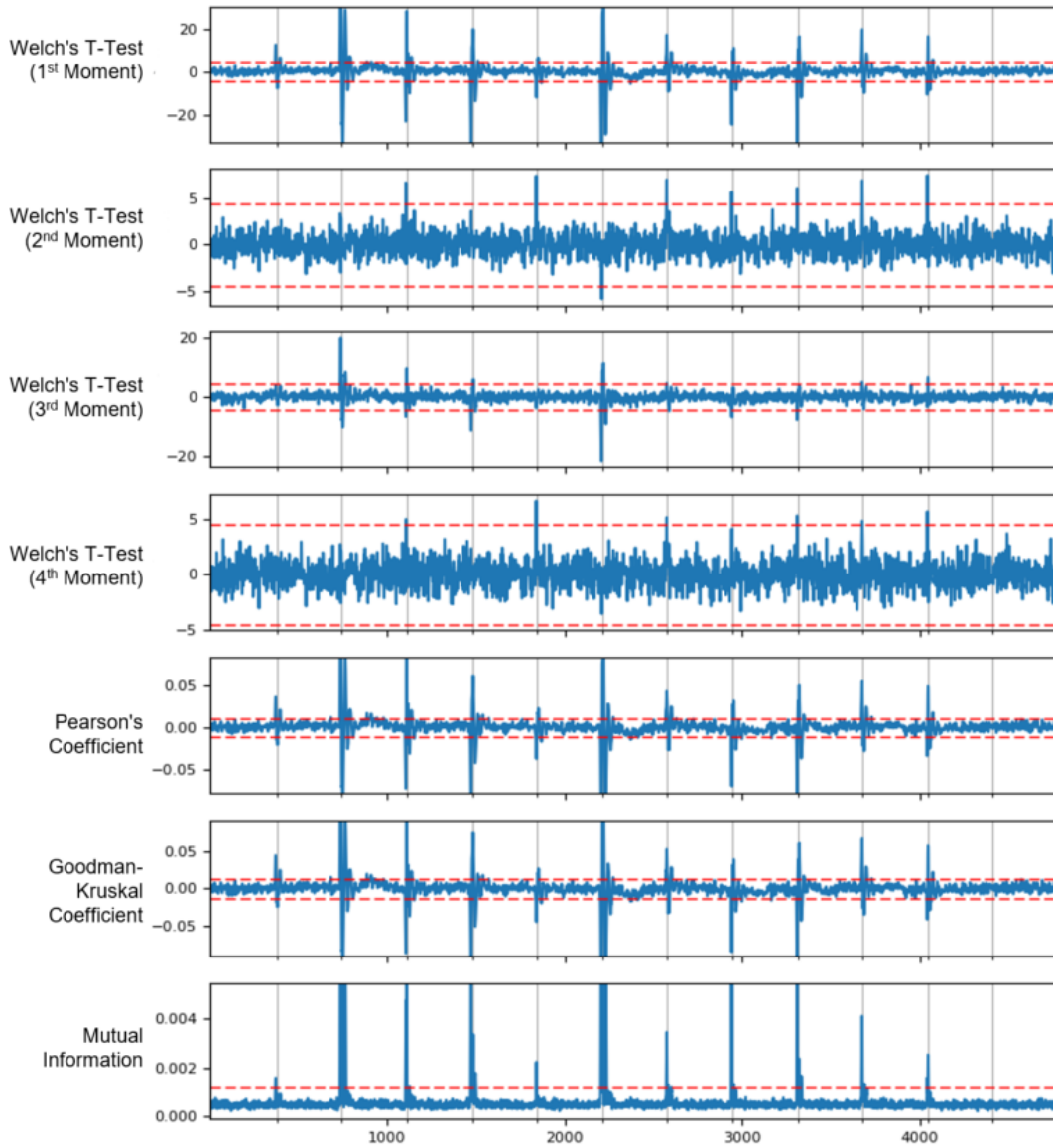


Figure 4.16: Zooming in on results of power side-channel risk assessment with 100,000 traces.

5 Conclusion

From the results discussed in the previous section, it appears mutual information could serve an important role in power side-channel risk assessment. The clear advantage is that mutual information identifies all types of leakage without relying on statistical moment estimation or assuming other adversarial strategies. However, it still may not be suitable for all situations. Statistical moment targeting tests will identify leakage with fewer traces than the more universal mutual information analysis. If trace collection time is an issue, this may be an important factor. However, there seems to be a point where mutual information begins to yield more statistically significant results. At these higher population sizes, mutual information's precision increases to a level that is unattainable by the t-test.

This crossing point is greatly dependent on the signal-to-noise ratio of the set of observations. This was seen on multiple occasions. In the unit test which varied SNR of multiple distributions, it was clear that mutual information yielded more significant results until noise was added. At the SNR in which the mutual information statistic became insignificant, the t-test was still above its threshold. Also, looking back at the simulated AES results, it took very little traces for mutual information to surpass the precision of other tests. This was the case in all simulation results because noise is not as prevalent. So, in the case of simulation, less traces would be required to see very confident leakage hits. For power trace results, however, noise is very strong compared to the signal of interest. At 10,000 traces, we saw that the t-test identified every AES round and MI only identified a few. But at 100,000 traces, the significance of MI at each round surpassed that of the t-test. From these tests, it is conclusive that moment targeting analyses identify sources of risk in fewer hits, but with more traces one can see far more precise results mutual information. If this level of precision and leakage generality is desired, mutual information could be a solution. Otherwise, in most situations the t-test will yield adequate leakage hits in fewer traces.

References

- [1] A. Ding, C. Chen, and T. Eisenbarth, "Simpler faster and more robust t-test based leakage detection," 2016.
- [2] G. Goodwill, B. Jun, and J. Jaffe, "A testing methodology for side-channel resistance validation," 2011. [Online]. Retrieved from: <https://42xtjqm0qj0382ac91ye9exr-wpengine.netdna-ssl.com/wp-content/uploads/2015/08/a-testing-methodology-for-side-channel-resistance-validation.pdf>
- [3] Gierlichs, B., Batina, L., & Tuyls, P. (2007). Mutual Information Analysis: A Universal Differential Side-Channel Attack. Retrieved from <https://eprint.iacr.org/2007/198.pdf>
- [4] Standaert, F. (2017). How (not) to Use Welch's T-test in Side-Channel Security Evaluations. Retrieved from <https://eprint.iacr.org/2017/138.pdf>
- [5] Brier, E., Clavier, C., & Olivier, F. (2004). Correlation Power Analysis with a Leakage Model. Retrieved from <https://www.iacr.org/archive/ches2004/31560016/31560016.pdf>
- [6] Goodman, L., & Kruskal, W. (1954). Measures of Association for Cross Classifications. Retrieved from http://www.nssl.noaa.gov/users/brooks/public_html/feda/papers/goodmankruskal1.pdf
- [7] Kendall's Rank Correlation. (2000). Retrieved from http://www.statsdirect.com/help/nonparametric_methods/kendall_correlation.htm
- [8] StatisticsSolutions. (n.d.). Kendall's Tau and Spearman's Rank Correlation Coefficient. Retrieved from <http://www.statisticssolutions.com/kendalls-tau-and-spearman-rank-correlation-coefficient/>
- [9] Freedman, D., & Diaconis, P. (1981). On the histogram as a density estimator: L2 theory. Retrieved from <https://rd.springer.com/article/10.1007/BF01025868>
- [10] Tyagi, A. (2012, June 25). T-Private logic synthesis on FPGAs. Retrieved from <http://ieeexplore.ieee.org/document/6224321/#full-text-section>