A testing methodology for side-channel resistance validation

Gilbert Goodwill, Benjamin Jun, Josh Jaffe, Pankaj Rohatgi: Cryptography Research Inc.

Keywords: side-channel testing, leakage analysis, t-test

1. Introduction

The goal of a side-channel resistance validation program is to assess whether a cryptographic module utilizing side-channel analysis countermeasures can provide resistance to these attacks commensurate with the desired security level. While, no standardized testing program can guarantee resistance against all attacks, an effective program should be able to validate that sufficient care was taken in the design and implementation of countermeasures. An effective validation program should be based on the following requirements:

- Effectiveness of tests: results should be reproducible and be reasonable indicators of resistance achieved
- Ease and cost-effectiveness of testing: Validating a moderate level of resistance (e.g., FIPS 140 Level 2/3 [1,2]) should not require excessive amount of testing time per algorithm or exceptional test operator skills

This paper proposes a testing methodology for side channel resistance validation that meets these requirements. The approach is based on the following rationale: Side-channel attacks such as SPA and DPA [3] exploit the presence of information about sensitive algorithmic intermediates within the side-channel traces collected from a device. Any sensitive computational intermediate that influences the side-channel in a statistically significant way could potentially create vulnerabilities. The approach uses statistical hypothesis testing to detect if one of a number of sensitive intermediates significantly influences the measurement data. For each sensitive intermediate, the collected traces are partitioned into two sets where the value of the intermediate is substantially different. The null hypothesis is that the two sets of power traces have identical means and variance. In other words, sensitive intermediate has no influence on these quantities. The alternate hypothesis is that the means of the two distributions is different. The Welch's t-test used in the proposal determines whether a data set with a size comparable to the acquired data set of an attacker, provides sufficient evidence to reject the null hypothesis. Note that we do not require full key extraction to fail a device: in our methodology, a device can fail if significant sensitive information leakage can be demonstrated. The same methodology would apply to test for higher-order leakages – in this case, the traces would require pre-processed before statistical testing.

The proposal is based on CAVP and more than a decade of side-channel testing practice by the authors. Labs collect specified measurement data from the device and apply a suite of statistical significance tests on the collected data. Our methodology has several advantages over alternate approaches that involve testers performing a battery of side-channel attacks in an attempt to recover the key:

- Simplification and standardization of the test process:
    - Side-channel measurements are captured using standardized test vectors and process
    - A standard statistical test is applied to detect different types of leakage
- Separating the testing methodology from the ever-evolving set of side-channel attacks:
    - Operator does not need to be proficient in an evolving set of side-channel attacks
    - As newer side-channel attacks emerge that exploit leakages not covered by the existing test-suite, the DTR authors can revise the test vectors and the datasets used for testing and adjust the

significance thresholds to capture these leakages. The basic statistical technique remains the same

We propose a specific set of test vectors and tests for validating that AES implementations against first order attacks – such testing would be appropriate for devices requiring moderate level of resistance to side-channel attacks. We then experimentally evaluated this testing methodology against three different AES implementations. In all cases, consistent and repeatable results were obtained within the six hours testing goal. For failing devices, the time to obtain repeatable results was often considerably less

The rest of the paper is organized as follows: Section 2 provides an overview of our testing methodology, provides guidelines for data collection and side-channel measurements, and provides a justification for the use of the Welch's t-test and our recommended significance thresholds. Section 3 provides a detailed set of test vectors for testing AES implementation to moderate level of resistance and provides some of the design principles behind the choice of these specific test vectors. In Section 4, we present our experimental results on using this methodology for AES testing and we conclude in Section 5.

2. Proposed Testing Methodology: Overview

The proposed methodology relies both on the review of vendors supplied documentation as well as well as side-channel measurements and tests performed on of the device.

Required Vendor Information

The vendor must provide the following documentation regarding the algorithms and countermeasures implemented within the device:

1. Implemented cryptographic algorithms.
2. Design of the implementation.
3. Vendor analysis of which algorithm and mode(s) of usage are susceptible to side-channel analysis, as well as specific side-channel exposures for these algorithms.
4. Any limits placed by the device on the number of side-channel samples available per secret key to an attacker.
5. Countermeasures adopted within the implementation.
6. Rationale for countermeasures and why countermeasures cover the exposures identified in number 3
7. Supporting documentation to validate that the claimed countermeasures are activated upon performing relevant cryptographic operations. This could include specification of the device state machine illustrating specific countermeasure activation upon initiation of a cryptographic operation

The vendor should also provide a test device and fixture for performing test operations and measuring power consumption. This test device must allow the tester to invoke the cryptographic operations with the test vectors, but otherwise be identical to the final product.

Summary of Test Procedures

The tester will verify that the side-channel exposures identified by the vendor are complete and that the identified exposures and the coverage provided by the countermeasures are consistent. This includes verifying that:

1. The vendor's assertions about which cryptographic algorithms and modes are subject to side-channel attacks are correct and complete based on device usage and lifecycle.
2. The rationale about coverage provided by the countermeasures is correct.

3. The appropriate set of countermeasures is active or activated whenever a vulnerable cryptographic operation is performed.

The tester then determines the number of measurement traces that should be collected from the device during testing. This is done using the following guidelines:

1. The DTR shall specify a default minimum number of measurement traces and a specified amount of time that a reasonable effort of data collection should take for the target security level.
2. If the tester determines that the device does not limit the number of traces available to an attacker for a single key, the tester shall determine the number of measurement traces to be collected to be the greater of the minimum number specified in the DTR and the number of traces that may reasonably be collected from the device within the time specified by the DTR.
3. If the device limits the number of side-channel traces available to an attacker per secret key, the tester shall apply that limit to the number of traces collected for testing.

The tester then collects the required number of side-channel traces from the DUT using the test vectors specified by the DTR and the guidelines given below. These traces are aligned and any preprocessing steps specified in the DTR are then performed. Finally, the tester performs the statistical tests for leakage on the aligned and/or preprocessed traces as specified in the DTR. Finally the tester uses the statistical test pass/fail criteria for each test specified in the DTR to determine whether the device has passed or failed – the failure of any single statistical test of leakage should be sufficient to fail the device.

We now provide details of the data collection process and provide details and justifications for the statistical tests and pass/fail criteria.

Data Collection

Cryptographic Parameters and Data

For each cryptographic algorithm, the DTR document will state a specific key or set of keys and a set of input data upon which the algorithm shall operate.

Measuring Power Consumption

As shown in the Figure 1 below, power must be measured by having an external power-supply attached to a resistor in series with the VCC line supplying the DUT and measuring the voltage drop across the resistor R, (between A and B) or the voltage between B and the ground (after removing the DC bias).

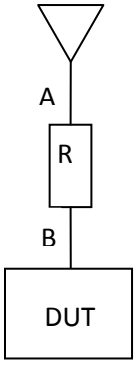

Figure 1: Measuring power consumption

If multiple power lines supply the device, select the line that is most likely to contribute to the powering of the cryptographic algorithm, based on vendor documentation. The tester shall choose the power line used for measurement. Table 1 provides example settings for the value of resistor and external bench voltage for a variety of example device characteristics.

Example Device Category	Specified Nominal Voltage	Max. Working Tolerance (%)	Example Current Range	Suggested Resistance	Suggested Input Voltage
Smart Card	5.0V	+/-10%	10-110mA	10 Ohm	5.60V
uController	3.3V	+/-10%	50-144mA	7 Ohm	3.98V
AES Core	1.9V	+/-5%	8-17mA	20 Ohm	2.14V
Small FPGA	1.2V	+/-10%	330-800mA	0.5 Ohm	1.49V
Medium FPGA	1.0V	+/-8%	800-1330mA	0.3 Ohm	1.32V

Table 1: Example DUT power measurement settings

Note the following:

- Voltage tolerance is the maximum range that a device can tolerate without exceeding actual working range at >95% reliability. The working range is typically wider than the tolerance listed in the specification sheet
- Input voltage is the voltage provided by the test harness "upstream" of the DPA resistor

In general, place the highest value resistor (up to 20 ohm) that allows the device to function. A current probe, or a near-field magnetic probe can be used in lieu of the resistor, provided the bandwidth of the probe is at least that of the device clock rate. A ground-side measurement can also be used if convenient.

DC-blocking or adding a DC offset is recommended when capturing data using a resistor. The information used for side-channel attacks is based on detectable variations in current/power-consumption, rather than its absolute calibrated value. The goal is to capture the variations in current or power consumption using the full dynamic range of the scope or sampling card.

Requirements for Measurement Apparatus

The power consumption measurements of the device while it performs cryptographic operations are captured on a sampling oscilloscope or A/D sampling card. The following are minimum requirements for the oscilloscope or A/D sampling cards to be used:

1. Bandwidth of at least 50% of the device clock rate for software implementations and at least 80% of the clock rate for hardware implementations
2. Capability to capture samples at 5x the bandwidth
3. A minimum of 8-bits of sampling resolution
4. Enough storage to capture the entire signal required for the test and analysis

Requirements for Data Collection

The following are requirements for data collection:

1. For each individual cryptographic operation that was part of the test, the measurement process shall record the key used, the input data, and the result produced by the algorithm implementation in the DUT. The measurement process must also specify where the actual measurement trace for the operation can

be found. During testing, there is a possibility that the device may malfunction, so the result must to be checked against the algorithm for each trace. Metadata about the data collected may be stored in a file, with one line for each trace

2. The signal must be amplified or the gain setting on the scope or sampling card must be set to measure the full dynamic range (or as much as possible) of the signal being collected
3. The sampling rate can be set based on test operator experience. At a minimum, the rate is set to 5x the bandwidth, with an input bandwidth close to the device clock frequency
4. The scope or sampling card must be triggered so that all parts of the cryptographic operation required by the test are captured
5. A measurement trace must consist of an optional, well-specified fixed length header followed by the raw measurements. The measurements should start with the beginning of the cryptographic operation or the specific location in the computation specified by the test. In case the measurements contain extra data, the actual start of the relevant measurement data must be specified. For example, include this information in the metadata record for each operation. One way to establish the start of the start of the relevant data is to have the implementation generate a trigger signal when the cryptographic operation is started and storing the traces corresponding to the trigger

Signal Processing

Signal alignment

When performing side-channel analysis, it is common to perform signal alignment so that different traces can be compared at the same point during the cryptographic calculation. For the purposes of side-channel testing, the vendor is required to cooperate with the testing lab to provide the best synchronization signal for the start of the cryptographic operation. For example, in testing mode the device may provide an external trigger point to indicate the start or stop of the cryptographic operation. If such start and stop information is not available, the testing lab must adopt standard signal processing- and matching-based techniques to perform alignment. In case the vendor is unable to provide well aligned measurements, the testing lab may increase the number of traces used for the test to compensate for additional noise introduced due to misalignment. In cases where traces are well aligned at the start of the cryptographic operation, the lab may be required to use standard, least-squares fit-based signal matching to perform better alignment on specific internals of the algorithm – the number and locations of these alignment points will be specified in the DTR. In this case, the lab may not collect additional traces, but the DTR may specify that testing be performed using multiple trace alignments.

Signal Preprocessing

The DTR can specify further processing of the aligned traces. For example, for a higher-order analysis, the DTR may specify that a new set of traces be created by applying a function on the aligned traces.

Leakage testing

The data collection and signal processing creates a set of measurement traces and metadata such as key, input and output for each trace. The DTR lists the statistical tests to apply to the data. The core statistical test and the pass or fail criteria are identical across all these tests. The pass and fail criteria can be the same for different algorithms. The tests differ on what classes or subsets of measurement traces are compared for statistically significant differences.

Statistical test and justification

The data collection stage must yield a collection of power traces corresponding to a set of cryptographic operations performed according to the DTR. Each statistical test is designed to measure the influence of a specific

kind of sensitive information on the power trace. Accordingly, the DTR will specify how the traces are to be divided into two subsets such that the sensitive information being processed is significantly different between the two subsets. This partitioning into two subsets need not be based on an specific intermediate bit, so this is similar to the "*generalized DPA*" approach in [4]. Partitioning of data into subsets based on sensitive algorithmic information is feasible since the cryptographic algorithms are performed with known parameters and data, and all intermediate states are known. Some algorithms may require a randomly generated parameter. In such cases, the DUT must allow this parameter to be set or known externally.

If the power traces in the two subsets are statistically different with high confidence, then information leakage is present and the device fails. Otherwise information leakage is either not present or is suppressed.

The core statistical technique for checking for statistical differences between the two subsets of power traces is the Welch t-test, which is an extension of the student t-test for unequal sample sizes and unequal variance. A high positive or negative value of T at a point in time, indicates a high degree of confidence that the null hypothesis is incorrect. The confidence value C will be specified in the DTR, and will correspond to a high confidence in rejecting the null hypothesis. C is chosen such that the probability of the t-statistic being greater than C or less than –C may correspond to 95%, 99% or even 99.99999% confidence that the null hypothesis can be rejected.

Repeating the test: Each trace can contain several thousand power measurements across time. Therefore, even for a fairly high threshold of C, chosen to make the likelihood of a false positive at a particular point in time small, there could be a significant likelihood that the t-test statistic exceeds +/- C at some point for large traces. To balance the need for detecting leakages (by keeping C small) while minimizing false positives, two independent experiments are required, and a device can be rejected only if the t-test statistic exceeds +/- C at the same time, in the same direction, in both experiments. If a particular leakage of information occurs at a particular point in the traces, then it should appear in both tests, whereas if the t-test statistic exceeded +/- C at a particular instance in time purely by chance, this rare occurrence is unlikely to repeat at the same instance in time in the next independent experiment.

For each algorithm, multiple t-tests must be performed, each targeting a different type of leakage. Each test must be repeated twice, with two different data sets.

Each test is performed as follows:

1. Specification of data to be used: The DTR specifies what the set of traces must be used for the test. The set of traces are divided into two disjoint groups, Group 1 and Group 2. These are the two disjoint data sets for performing the two independent Welch t-tests.
2. Group 1 test:
    a. The DTR for the algorithm specifies a partitioning the traces in Group 1 into two subsets A and B. Let N_A and N_B be the size of the subsets A and B.
    b. Compute X_A the *average* of all the traces in group A, X_B the *average* of all traces in group B, S_A the *sample standard deviation* of all the traces in group A and S_B, the *sample standard deviation* of all the traces in group B. Note that, as each trace is a vector of measurements across time, and the average and sample standard deviations of the traces are also vectors over the same points in time, i.e., the averages and sample standard deviations are computed point-wise within the traces for each point in time.

c. Compute the t-statistic trace T (over the same time instants) as $\frac{X_A - X_B}{\sqrt{\frac{S_A^2}{N_A} + \frac{S_B^2}{N_B}}}$

Note that the above calculation is performed point-wise, for each time instant in the traces for X_A, X_B, S_A and S_B.

d. Note the time instants in the t-test statistic trace T, where the value exceeds the confidence threshold +/- C (where C is specified in the DTR).

3. Group 2 test: The testing steps are the same as those for Group 1, except that the measurement traces, and its subsets A and B will be different. Again the time instants where the t-static trace computed over Group 2, exceeds the threshold +/- C are noted.

Pass/Fail Criteria

If there is any point in time for which the t-test statistic exceeds + C for both Group 1 and Group 2 or is below – C for both group, the device fails. Otherwise, the device passes this test. In some cases, the DTR may specify a particular region in the trace to consider for a particular test. For example, a test may require that only the time-instances corresponding to the middle third of the AES calculation or an RSA exponentiation be used to determine pass/fail.

Related work

The topic of choice of statistical tests and distinguishers for side-channel attacks has received much attention in the published literature. Starting from "difference of means" based introduced in the original DPA paper [3], several techniques such as all-or-nothing DPA, multi-bit DPA and generalized DPA [4], maximum-likelihood based DPA [5], correlation power analysis [6], mutual Information based distinguisher [7], stochastic methods[8], template attacks[9] etc, have been proposed. The focus in such tests has been distinguishing the correct key from incorrect key guesses, in an attack setting. While our approach, which is based on detection of information leakage in the side-channel is more fundamental than the distinguishing the correct key from the incorrect guesses, some of the results from research on distinguishers is applicable to our setting. In particular, when considering DPA-style (univariate) side-channel attacks, as would be appropriate for validating moderate level of side-channel resistance, the research [10,11] indicates that the specific choice of technique does not have a major impact in the asymptotic sense. In particular, our proposed technique which is similar to Generalized DPA, but corrects for the variance, is known to be comparable to other approaches, provided the leakages being targeted actually exist in the device. The usage of the t-test in side-channel analysis has also been proposed in literature. For example, the formulae derived for maximum likelihood based DPA testing in [5] are quite close to the formulae for the t-test, these maximum likelihood formulae was also proposed for selecting relevant points for developing templates in [12] and the t-test was directly used for this purpose in [13].

Practical Considerations

The testing methodology described here can be easily automated. It is also possible to perform a large number of statistical tests for different leakages concurrently on the same data.

During data collection, the device can malfunction or trigger missed conditions. Set up data collection so that device output is checked and the collection process can recover from a missed trigger.

During data collection, it is common to have slight time changes in power measurements. This may result from changes in operating temperatures of the measurement equipment and the DUT. It is recommended that the data sets used for the statistical tests be collected in an interleaved manner, in order to remove systemic bias that can

create false positives when traces collected during one period of time are compared against traces collected during another. For specific algorithms, the DTR will identify cases where such an interleaved collection is recommended.

3. Testing AES at Moderate Security Level

For AES implementations to be validated at a moderate security level (e.g., FIPS 140 level 3), resistance against un-profiled, first-order attacks may be sufficient. The minimum number of operations should be 5,000 and a data collection time should be at least three hours.

AES Test Vectors

The tester must collect two data sets DATA-SET-1 and DATA-SET-2 from the core AES cryptographic block encryption with a specific, published key and a set of data as follows:

1. DATA-SET 1:
    a. Key K is set to
        i. 0x0123456789abcdef123456789abcdef0 for AES-128
        ii. 0x0123456789abcdef123456789abcdef023456789abcdef01 for AES-192
        iii. 0x0123456789abcdef123456789abcdef023456789abcdef013456789abcdef012 for AES-256
    b. Perform 2n encryptions with inputs: $I_0, I_1, \ldots, I_{2n}$
    Where I_0 = 0x00000000000000000000000000000000 (16 0 bytes)
    and $I_{j+1} = AES(K, I_j)$ for $0 < j < 2n$.
    where n is the number of distinct encryption samples chosen by the tester after the review of the documentation. The number of inputs and number of encryptions for this data set shall be twice the number of distinct samples deemed reasonable for an attacker to collect. In this case, the key K will perform 2n AES encryptions. The input for the first encryption shall be all zeros and each subsequent encryption uses the output of the previous encryption as its input.
2. DATA-SET 2:
    a. Key K is set to
        i. 0x0123456789abcdef123456789abcdef0 for AES-128
        ii. 0x0123456789abcdef123456789abcdef023456789abcdef01 for AES-192
        iii. 0x0123456789abcdef123456789abcdef023456789abcdef013456789abcdef012 for AES-256
    b. Data J is set to
        i. `0xda39a3ee5e6b4b0d3255bfef95601890 for AES-128`
        ii. `0xda39a3ee5e6b4b0d3255bfef95601888 for AES-192`
        iii. `0xda39a3ee5e6b4b0d3255bfef95601895 for AES-256`
    c. Perform n encryptions with input J

DATA-SET 2 uses the same key as DATA-SET 1, but repeatedly performs an encryption with a single fixed data value. Both DATA-SET 1 and DATA-SET 2 require the entire AES operation to be measured, recorded and checked.

Practical considerations: It is recommended that the collection of DATA-SET 2 be interspersed with the collection of the first n traces from DATA-SET 1. This is to eliminate any systemic time-dependent bias between traces from DATA-SET 1 and DATA-SET 2 used for subsequent testing.

Rationale

DATA-SET 1, was chosen to essentially provide a set of AES operations with a fixed key (for different key sizes) and a deterministic set of inputs which appear essentially random from a statistical perspective.

DATA-SET 2 was chosen to satisfy the following criterion: The choice of key was identical to DATA-SET 1. The data J was selected such that the following four conditions were met (for AES-128, AES-192, AES-256):

1. In one middle round (outside first and last round) there is at least one data byte equal to zero.
2. In one middle round, there is at least one byte of data XOR round key equal to zero.
3. In one middle round, there is at least one S output byte equal to zero.
4. In one middle round, there is at least one byte of round_in XOR round_out equal to zero.

To find such a J, the following approach was used: Starting with J = first 121 bits of SHA1("TEST0") followed by 7, "0" bits, increment J until the criterion is met.

Example Tests for AES

Based on proposed data collection for AES, the following are examples of tests that can be carried out:

1. The tester must choose a value for R (the round intermediates to be tested) without informing the vendor. Values for R must be one of the following:
    a. between 2 and 8 for AES-128
    b. between 2 and 10 for AES-192
    c. between 2 and 12 for AES-256
2. Test 0
    a. Group 1: First n/2 traces in DATA-SET 2, First n/2 traces in DATA-SET 1
    b. Group 2: Last n/2 traces in DATA-SET 2, n/2+1 through n traces in DATA-SET 1
    c. Subset A : traces from DATA-SET 2
    d. Subset B: traces from DATA-SET 1
    e. Region of interest for pass/fail: middle 1/3 of AES operation
3. In the following 896 tests (or 384 tests if n < 5000), the same data groupings are used:
    a. Group 1: Traces 1 through n from DATA-SET 1
    b. Group 2: Traces n+1 through 2n from DATA-SET 1
    c. The entire trace (all time-instances for the AES calculation) can be used for pass/fail determination; however, the test processing can be optimized if only the region of the AES trace at round N-1, N and N+1 are considered.
4. Tests RIRObitR_0 through RIRObitR_127:
    a. For each trace let RIRO_N = EXOR or Round R input with Round R output
    b. For RIRObitR_i (where i varies from 0 to 127)
        i. Subset A: traces where bit i of RIRO_R = 0
        ii. Subset B: traces where bit i of RIRO_R = 1
5. Tests SoutbitN_0 through SoutbitR_127:
    a. For each trace let Sout_R = Concatenated output of the 16 Sbox table lookups in Round R
    b. For SoutbitR_i (where i varies from 0 to 127)
        i. Subset A: traces where bit i of Sout_R = 0
        ii. Subset B: traces where bit i of Sout_R = 1
6. Tests RoutbitN_0 through RoutbitN_127:
    a. For each trace let Rout_N = Output of Round N

b. For RoutbitN_i (where i varies from 0 to 127)
    i. Subset A: traces where bit i of Rout_N = 0
    ii. Subset B: traces where bit i of Rout_N = 1

7. Tests Routbyte_N_0_0 through Routbyte_N_0_255: (to be carried out only when n > 5000)
    a. For each trace let Rout_0_N: First byte of output of Round N
    b. For RoutbyteN_0_i (where i varies from 0 to 255)
        i. Subset A: traces where Rout_0_N ≠ i
        ii. Subset B: traces where Rout_0_N = i
8. Tests Routbyte_N_1_0 through Routbyte_N_1_255: (to be carried out only when n > 5000)
    a. For each trace let Rout_1_N: Second byte of output of Round N
    b. For RoutbyteN_1_i (where i varies from 0 to 255)
        i. Subset A: traces where Rout_1_N ≠ i
        ii. Subset B: traces where Rout_1_N = i

Pass/fail criteria

For n> 100, the pass fail criteria is C > 4.5 or C < -4.5. For large n, (n > 5000) this implies a confidence of > 99.999% for Test 0, the first set of 384 tests that the null hypothesis does not hold and at least a confidence of 99.95% for the second set of 512 tests For 100 < n < 5000, e.g., in devices that employ a countermeasure to restrict the number of AES encryptions possible using the same key, only Test 0 and the 384 subsequent bit oriented tests must be performed. In these cases, the choice of C=4.5, corresponds to rejection of null hypothesis at confidence of to 99.99%.

For n < 100, the Welch correction to the degrees of freedom must be applied when evaluating the t-statistic. In these cases, the pass/fail criteria is the presence of a point in time, where the null hypothesis rejected with 99.5% confidence, in the two independent trials. Note that in these cases, the C value that corresponds to 99.5% confidence can no longer be assumed to be constant across all the time-instants, since it depends not only on the sizes of the subsets but also on the value of the sample standard deviations at each time-instant.

4. Experimental Results for AES Testing

To benchmark the testing proposal, we tested 3 AES based devices against the proposed testing methodology. Two of the DUTs had AES implementation without countermeasures and one DUT employed DPA countermeasures. For each DUT, a definitive result (PASS/FAIL) was obtained in less than 6 hours, which included time for data collection, processing, testing and interpretation. The data was collected for a fixed amount of time (~ 3 hours). Processing and testing was started while the traces were being collected and early exit was possible if the device failed with fewer traces.

There were essentially two classes of tests:

1. Fixed vs. random data test
2. Leakage tests for varying data. There are essentially 4 subclasses such leakage tests
    a. XOR of round input and output (AES 128 bits)
    b. S-box output for round (AES 128 bits)

c. Round output (AES 128 bits)
d. Byte analysis of round output
    i. Each value for first byte, second byte (2*256 values)

We now describe our experimental results for each of these DUTs.

DUT 1

DUT 1 (Figure 2) was an AES128 implementation in software on a smart card, running at 4 Mhz with no countermeasures.

Figure 2: DUT 1

Traces could be collected from the device at a rate of 12/second. Based on test methodology specification for 3 hours of data collection, 130,000 traces could be collected in 3 hours (n = 43000). However, by running the tests with partial data, concurrently with data collection, an early exit condition was reached within only 4 minutes of data collection.

DUT1: Test Results

Unsurprisingly,DUT1 was a **definitively FAIL.** The tester selected Round 6 for analysis. The device failed most tests (the failure of any single test sufficient to fail DUT). In fact, the only tests that the device did not fail were in category 2a (maximum t-statistic for these tests was 4.17 across all 128 bits) The leakage rate was high, as can expected for device without countermeasures. Table 2 gives the highest t-statistic values for each of category of tests.

Test Category	Maximum t-statistic across all tests in category
Fixed vs. Random	74.9
Round 6, XOR round input with round output bits	4.17 (bit 118)
Round 6, S-box output bits	26.9 (bit 63)
Round 6, Output bits	57.1 (bit 31)
Round 6 Output, 1st byte	62.7 (value 186)
Round 6 Output, 2nd byte	62.3 (value 172)

Table 2: Summary of test results for DUT 1

Figure 3 shows a failing t-test for DUT 1, showing the t-statistic exceed the +/- 4.5 threshold at several points in time for two independent data sets.

Figure 3:Failing t-test for DUT 1

The total testing time was 14 minutes, including 4 minutes for data collect and 10 minutes for analysis.

DUT 2

DUT 2 (Figure 4) is the sample AES128 implementation on the Sasebo-GII FPGA board. It operates at 24 MHz and contains no countermeasures.

Figure 4: DUT 2

Traces could be collected from the device at a rate of 20/second. Based on test methodology specification for 3 hours of data collection, 216,000 traces could be collected in 3 hours. However, by running the tests with partial data, concurrently with data collection, an early exit condition was reached within only 50 minutes of data collection.

DUT2: Test Results

Unsurprisingly, DUT2 was a **definitively FAIL.** The tester selected Round 4 for analysis. The device failed many tests (the failure of any single test sufficient to fail DUT). In fact, failed all tests in categories other than 2b and 2c (maximum t-statistic for these tests was 4.23 and 3.68 respectively across all 128 bits) The leakage rate was high, as can expected for device without countermeasures. Table 3 gives the highest t-statistic values for each of category of tests.

Test Category	Maximum t-statistic across all tests in category
Fixed vs. Random	190
Round 4, XOR round input with round output bits	24 (bit 19)
Round 4, S-box output bits	4.23 (bit 89)
Round 4, Output bits	3.68 (bit 41)
Round 4 Output, 1st byte	5.15(value 254)
Round 4 Output, 2nd byte	5.54 (value 173)

Table 3: Summary of test results for DUT 2

Figure 4 shows a failing t-test for DUT 2, showing the t-statistic exceed the + 4.5 threshold at the same time for two independent data sets.

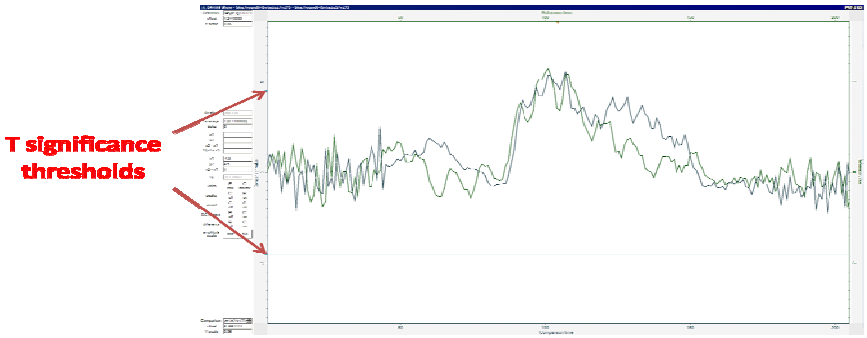

Figure 5: Failing t-test for DUT 2

The total testing time was 62 minutes, including 50 minutes for data collection and 12 minutes for analysis.

DUT 3

DUT 3 is a masking-based, DPA-protected AES implementation on a Sasebo-GII FPGA board, operating at 24 MHz.

Traces could be collected from the device at a rate of 20/second. Based on test methodology specification for 3 hours of data collection, 216,000 traces could be collected in 3 hours. The device usage mode did not permit it to perform repeated operations with the same key and data, so only DATA-SET 1 was collected and only tests from category 2 were conducted. Since the device passed all tests, a full 3 hours of data collect was required in this case.

DUT3: Test Results

DUT3 was a **definitively PASS.** The tester selected Round 7 for analysis. The device passed all tests in category 2. Table 4 gives the highest t-statistic values for each of category of tests.

Test Category	Maximum t-statistic across all tests in category
Fixed vs. Random	N/A
Round 7, XOR round input with round output bits	3.72 (bit 83)
Round 7, S-box output bits	3.47 (bit 43)
Round 7, Output bits	3.38 (bit 47)
Round 7 Output, 1st byte	3.94 (value 216)
Round 7 Output, 2nd byte	3.46 (value 85)

Table 4: Summary of test results for DUT 3

Figure 6 shows a typical result from the testing, showing t-statistic value across time for two independent data sets. As the figure shows, the t-statistic remained between the +/- 4.5 significance thresholds across time for both experiments.

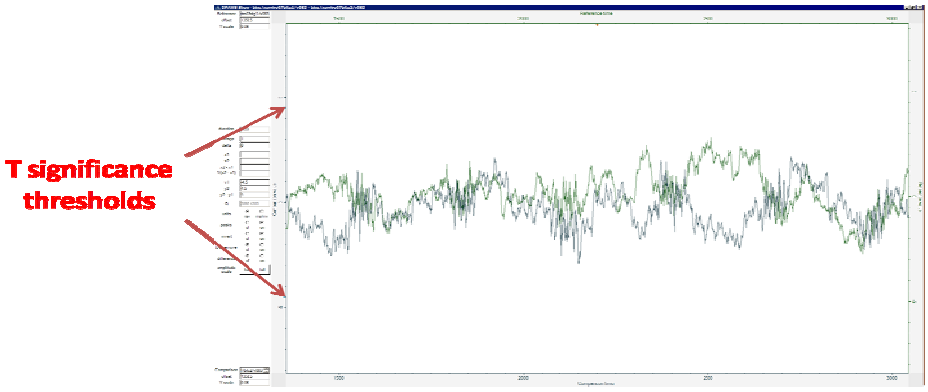

Figure 6: Results of typical t-test

The total testing time was 3 hours and 20 minutes which included 3 hours for data collect and 20 minutes for performing testing on Round 7 intermediates. Subsequently, testing was performed on all middle rounds and no leakage was found. Testing for all intermediate rounds minutes took 3 hours.

5. Conclusion

In this paper, we have a proposed a simple testing methodology suitable for side-channel resistance validation. It specifies a set of simple, repeatable tests with a clear pass/fail criterion. Our experimental results indicate that this testing methodology is effective in finding leakages that could result in side-channel vulnerabilities. The testing process can be automated, once the operator has established confidence in the data. The process is efficient and does not require high test operator expertise.

6. References

1. *FIPS PUB 140-2: Security Requirements for Cryptographic Modules*, http://csrc.nist.gov/groups/STM/cmvp/standards.html#02
2. *FIPS 140-3 DRAFT Security Requirements for Cryptographic Modules (Revised Draft)*, http://csrc.nist.gov/publications/drafts/fips140-3/revised-draft-fips140-3_PDF-zip_document-annexA-to-annexG.zip
3. Kocher, P., Jaffe, J., Jun, B.: *Differential Power Analysis*. In Weiner, M., ed: Advances in Cryptology – CRYPTO '99. Volume 1666 of LNCS., Springer, 1999, pages 388-397.
4. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: *Examining smart-card security under the threat of power analysis attacks.* IEEE Trans. Computers **51**(5) (2002) pages 541–552.
5. Agrawal, D., Rao J.R., and Rohatgi P.: *Multi-channel Attacks*. CHES 2003, volume 2779, Springer-Verlag, 2003, pages 2–16.
6. Brier, E., Clavier, C., Olivier, F.: *Correlation Power Analysis with a Leakage Model*, CHES 2004, LNCS, vol 3156, Boston, MA, USA, August 2004, pages16-29.
7. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B., *Mutual Information Analysis*, CHES 2008, LNCS, vol 5154, Washington DC, USA, 2008, pages 426-442.
8. Schindler, W., Lemke, K., Paar, C.: A *Stochastic Model for Differential Side-Channel Cryptanalysis*, CHES 2005, Lecture Notes in Computer Science, vol 3659, 2005, pages 30-46.
9. Chari, S., Rao, J.R., Rohatgi, P.: *Template Attacks*, CHES 2002, Lecture Notes in Computer Science, vol 2523, Springer, pages 13-28.
10. Mangard, S., Oswald, E., Standaert, F.-X.: *All for one-one for all: Unifying univariate DPA attacks*. IET Information Security, 5(2), July 2011, pages 100-110.
11. Julien Doget J., Prouff, E., Rivain, M., Standaert, F.-X.: *Univariate Side Channel Attacks and Leakage Modeling*, Cryptology ePrint Archives, http://eprint.iacr.org/2011/302.pdf
12. Agrawal, D., Rao, J.R., Rohatgi, P., Schramm, K.: *Templates as Master Keys*, CHES 2005, Lecture Notes in Computer Science, vol. 3659, Springer 2005, pages 15-29.
13. Gierlichs, B., Lemke-Rust, K., Paar, C.: *Templates vs. Stochastic Methods: A Performance Analysis for Side Channel Cryptanalysis.* CHES 2006 , Lecture Notes in Computer Science, vol 4249, 2006, pages 15-29.