Test Vector Leakage Assessment (TVLA) Derived Test Requirements (DTR) with AES

1 Document Scope

This document describes requirements and test procedures for qualifying DPA-resistant implementations of cryptographic algorithms with specific instructions and test vectors for AES. Expected lab and analyst proficiency, device setup, data acquisition, signal processing, analysis and evaluation procedures are described herein.

2 Lab/Analyst Qualifications

The tester shall have shown proficiency in side-channel analysis by having located leakage on provided test artifacts.

3 Device and Data Acquisition Set-up

The analyst shall perform device signal tapping, data acquisition, and post-data collection signal processing sufficient to demonstrate leakage of unmasked cryptographic quantities, such as input and/or output messages. Leakage from these values must be used to confirm successful signal tapping (for power) or probing (for electromagnetic analysis) and acquisition.

The device under test, preferably without shielding, should allow probing or power tapping in proximity to the where cryptographic operations are performed. Note capacitors near the ASIC may provide quality signal, despite being further from the cryptographic core performing the operations.

Data acquisition equipment for initial exploration should provide the ability to see frequencies at several multiples of the clock frequency, to provide sub-clock cycle resolution. Actual leakage may be, and often is, found at lower sub-clock frequencies. Standard considerations for sampling rates in regard to Nyquist frequency and filtering roll off rates should be taken into account. Amplification of signals may also be necessary.

Data acquisition components (probes, filters, amplifiers, tuners, receivers, etc.) must all be documented sufficiently to allow another party to recreate the results.

When setting the key, if the key is to be provided as separate shares, the key shares must use sufficient randomness to avoid leakage in General Test: Fixed-vs.-Random Key.

3.1 Requirements for Data Collection

The following are requirements for data collection:

1. For each input data, 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**.**
2. During testing, there is a possibility that the device may malfunction, so the result must to be checked against the algorithm for each trace.
3. 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.
4. The sampling rate can be set based on test operator experience. As a starting point, the rate is set to 5 times the bandwidth, with an input bandwidth at least twice the device clock frequency.
5. The scope or sampling card must be triggered so that all parts of the cryptographic operation required by the test are captured.
6. Known leaking quantities such as input, output or unmasked values, may be used to confirm and refine the setting listed above.

3.2 Post-data Collection Signal Processing

Following data collection, some amount of signal processing of the traces is likely to be necessary before evaluating side-channel leakage. Typically alignment is done with least squares fitting adjusting the offset of each trace relative to reference trace. For traces involving multiple transactions, the reference template transaction should be created, and offsets for each transaction within the traces should be located.

Where possible, the device shall provide a trigger signal indicating the start (and possibly stop) of the cryptographic operation, to aid in alignment.

If repeated operations differ in the amount of time they take, techniques such as clock compression or elastic alignment should be applied to resynchronize the traces in order to find leakage. Alternatively, the traces may be aligned at various different points (multi-point alignment) to target different intermediates.

Again, verification of proper signal processing should be done by making sure unprotected quantities such as input, output, or unmasked quantities show leakage.

Higher-order analysis (see 7.4 Second-Order Tests below) may require a form of signal pre-processing, applying a function to the aligned traces.

4 Selection of Round

The device may be tested for leakage of a middle round of a cipher, if all rounds of the cipher leak equivalently (e.g., all re-use the same hardware, supplying cipher state and round key). Otherwise, all rounds should be tested. Selection of a round for analysis, if appropriate, may also be used to restrict the period of time for data collection.

5 Datasets

Two classes of tests are included: general and specific. The general tests look for any leakage that depends on input data or key. Specific tests target specific intermediates of the cryptographic operation that could be exploited to recover keys or other sensitive security parameters. A failing general test shows leakage that may or may not be immediately exploitable, but indicates the possibility of exploit. Specific test failures indicate leakage that is directly exploitable for recovery of secrets.

For each of the pairs of datasets to be compared against each other in the 6 Welch's t-tests described below, traces from the two datasets should be collected randomly interspersed to avoid any systematic bias due to order of collection.

As indicated below all tests comparing datasets will be performed on twice, on independent data sets. (See 6 Welch's t-test.)

For any fixed-vs.-random set described below, care must be taken to avoid any systematic differences in setting up the device. If as part of the test infrastructure, the device, for example, uses different memory addresses depending on whether the fixed or varying key/data is being used, this may result in failing tests that are not due to the implementation itself. All device configuration and control flow on the device should be identical for both fixed and random inputs.

5.1 General Test: Fixed-vs.-Random Data Datasets

For the general test of fixed-vs.-random data, the key on the device, K_{dev}, is set to

- 0x0123456789abcdef123456789abcdef0 for AES-128
- 0x0123456789abcdef123456789abcdef023456789abcdef01 for AES-192
- 0x0123456789abcdef123456789abcdef023456789abcdef013456789abcdef012 for AES-256

For the general test of fixed-vs.-random data, for both fixed and random data datasets the generation key K_{gen} is set to

- 0x123456789abcdef123456789abcde0f0 for AES-128
- 0x123456789abcdef123456789abcdef023456789abcde0f01 for AES-192
- 0x123456789abcdef123456789abcdef023456789abcdef013456789abcde0f012 for AES-256

For the fixed data datasets, perform *n* encryptions of the input I_{fixed} set to

- 0xda39a3ee5e6b4b0d3255bfef95601890 for AES-128
- 0xda39a3ee5e6b4b0d3255bfef95601888 for AES-192
- 0xda39a3ee5e6b4b0d3255bfef95601895 for AES-256

For the random data datasets, perform *n* encryptions with the input set as follows

- I_0 = 0x00000000000000000000000000000000 (16 0 bytes)
- $I_{j+1} = AES(K_{gen}, I_j)$ for $0 \leq j < n$

5.2 Specific Tests: Random-vs.-Random Data Datasets

For the specific tests targeting intermediates (see 7.2 Specific Intermediates below) collect *2n* traces as described above for the random data portion of the Fixed-vs.-Random Test Dataset. The *n* traces collected there may be used as half of this dataset.

5.3 General Test: Fixed-vs.-Random Key Datasets

For the general test comparing fixed key to random key across datasets with random data, the datasets should be constructed as follows.

A fixed key data set with fixed key and varying data

- The fixed key K_{fixed} set on the device to
  - 0x811E3731B0120A7842781E22B25CDDF9 for AES-128
  - 0x811E3731B0120A7842781E22B25CDDF994F4D92CD2FAE645 for AES-192
  - 0x811E3731B0120A7842781E22B25CDDF994F4D92CD2FAE64537B940EA5E1AF112 for AES-256
- *n* varying plaintexts, I_j, generated as follows
  - I_0 = 0xAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  - I_{j+1} = AES(K_{gen}, I_j) for $0 \le j < n$

A data set with varying keys and varying data

- *n* varying keys, K_j, set on the device, generated as follows
  - S_0 = 0x53535353535353535353535353535353
  - S_{i+1} = AES(K_{gen}, S_i), using AES-128, -192, or -256 to match the cipher under test
  - For AES-128, take the S_j values directly as the keys
    - $K_j = S_j$ for $0 \le j < n$
  - For AES-192, concatenate pairs of S_j values, truncating to 192 bits, as the keys
    - $K_j = [S_{2j} \,||\, S_{2j+1}[0,63]]$ for $0 \le j < n$
  - For AES-256, concatenate pairs of S_j values as the keys
    - $K_j = [S_{2j} \,||\, S_{2j+1}]$ for $0 \le j < n$
- *n* varying plaintexts, I_j, generated as follows
  - I_0 = 0xCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  - I_{j+1} = AES(K_{gen}, I_j) for $0 \le j < n$

5.4 Semi-Fixed-vs.-Random Data

If the Fixed-vs.-Random Data test fails, it may be due to false-positive cases where non-cryptographic operations involving the inputs and/or outputs of fixed-vs.-random data overlap with the cryptographic operations on those quantities. will delay work?

If Fixed-vs.-Random Data failures occur and all rounds can be considered to have equivalent leakage (see 4 Selection of Round), an internally-semi-fixed dataset shall be collected. Data for this "semi-fixed-vs.-random" (SFVR) test shall be constructed as follows:

- Choose a middle round of the cipher (e.g., round 8 for AES-256)
- Set the cipher's state at the start of that round to the fixed SFVR round state value
    - 0x8B8A490BDF7C00BDD7E6066C61002412
- Vary the first 32-bits of the state as to create up to 2^{32} different values
- Set on the device the key K_{fixed} specified in 5.1 General Test: Fixed-vs.-Random Key Datasets for the fixed key dataset
- Invert the AES rounds on this state to compute the input that would create this state at the chosen round.

For the random data datasets, perform *n* encryptions with key K_{fixed} and with the input set as follows

- I_0 = 0x00000000000000000000000000000000 (16 0 bytes)
- $I_{j+1} = AES(K_{gen}, I_j)$ for $0 \leq j < n$

6 Welch's t-test

To evaluate leakage, the tests described here will be comparing two datasets, possibly of differing sizes and standard deviations, to see if they show statistically significant differences. Welch's t-test is to be used to evaluate leakages.

The Welch's t-test comparing the traces from subsets A and B is computed as follows:

a. 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.

b. 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.

Using Welch's t-test, a threshold of 4.5 standard deviations, corresponding to 99.999% confidence that a difference shown is not due to random change, will be set. Further, for each test, the t-test should be run twice on independent data sets. If for a given test the same time point in both data sets exceeds ±4.5σ, the device is leaking data-related sensitive parameters.

7 Individual tests

7.1 General Test: Fixed-vs.-Random Data

Partition the traces described in 5.1 General Test: Fixed-vs.-Random Data Datasets and compare the values at each point in time. Apply the ±4.5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.

7.2 Specific Intermediates

For AES, the following intermediates shall have their bits evaluated for leakage

- S-box outputs (128 bits)
- round outputs (128 bits)
- XOR of round input and round output (128 bits)

The following quantities shall have the byte values of their first 2 bytes evaluated for leakage

- S-box outputs (2 times 256 values)
- round outputs (2 times 256 values)
- XOR of round input and round output (2 times 256 values)

These “byte-wise” tests compare traces with a specific value in the targeted intermediate’s targeted byte against all other traces.

Partition the traces described in 5.2 Specific Tests: Random-vs.-Random Data Datasets by their intermediate bit and byte values as specified above and compare the values at each point in time. Apply the ±4.5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.

7.3 General Test: Fixed-vs.-Random Key

Partition the traces described in 5.1 General Test: Fixed-vs.-Random Key Datasets based on whether they are from the fixed or random datasets, and compare the values at each point in time. Apply the ±4.5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.

7.4 Second-Order Tests

Definitions for quantities used in describing second-order tests:

- The *mean-corrected value* at a time point in a trace is defined as the value at that time point with the mean across all traces at that time point subtracted.
- The *centered product* is defined as the mean-corrected value at one time point times the mean-corrected value at some other time point.

Higher-order side-channel leakage (HO-DPA) shall be evaluated, by computing the same Welch's t-test on traces pre-processed to contain the centered product of each time point and all shifted time points

within the range(s) corresponding to the cryptographic operation. As with first order analysis, if rounds are using equivalent hardware, or leakage can be shown to be equivalent between rounds, this second-order analysis may be performed on subset of the time points comprising the cryptographic operation. I.e., for traces containing *m* time points, not all *m*-by-*m* points need be computed if times where targeted intermediates are manipulated can be clearly shown. In this case, a subset of the second-order computation may be computed.

For second-order tests, the threshold for failure of the t-tests increased to ±5σ. With this adjustment, testing similar to the first-order case for fixed-vs.-random data and specific intermediates is to be performed.

7.4.1 Second-Order General Test: Fixed-vs.-Random Data

Pre-process the traces described in 5.1 General Test: Fixed-vs.-Random Data Datasets as described in 7.4 Second-Order Tests, and partition these traces based on whether they are fixed and random, and compare the values at each point in time. Apply the ±5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.

7.4.2 Second-Order General Test: Semi-Fixed-vs.-Random Data

If the 7.4.1 Second-Order General Test: Fixed-vs.-Random Data fails, it may be due to false-positive cases where non-cryptographic operations involving the inputs and/or outputs of fixed-vs.-random data overlap with the cryptographic operations on those quantities.

If Fixed-vs.-Random Data failures occur and all rounds can be considered to have equivalent leakage (see 4 Selection of Round), an internally-semi-fixed dataset shall be collected. Pre-process the traces described in 5.4 Semi-Fixed-vs.-Random Data as described in 7.4 Second-Order Tests, and partition these traces based on whether they are semi-fixed and random, and compare the values at each point in time. Apply the ±5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.

7.4.3 Second-Order Specific Intermediates

For AES, the following intermediates shall have their bits evaluated for second-order leakage

- S-box outputs (128 bits)
- round outputs (128 bits)
- XOR of round input and round output (128 bits)

The following quantities shall have the byte values of their first 2 bytes evaluated for second-order leakage

- S-box outputs (2 times 256 values)
- round outputs (2 times 256 values)
- XOR of round input and round output (2 times 256 values)

These “byte-wise” tests compare traces with a specific value in the targeted intermediate’s targeted byte against all other traces.

Pre-process the traces described in 5.2 Specific Tests: Random-vs.-Random Data Datasets as described in 7.4 Second-Order Tests, and partition them by their intermediate bit and byte values as specified above and compare the values at each point in time. Apply the ±5σ criteria across two independent test sets. Any failure at the same time point in both datasets indicates a failure for this test.