Radboud University

Test Vector Leakage Assesment (TVLA)

(AI-assisted) side-channel attacks on real-world crypto implementations

Hardwear.io 2024 (Amsterdam) training, Lecture 2, Day 2

Lejla Batina[1], Łukasz Chmielewski[2] & Péter Horváth[1]

October 22, 2024

[1] Digital Security group, Radboud University, The Netherlands
[2] Centre for Research on Cryptography and Security, Masaryk University, Czechia

Training schedule

- Day 1: Intro to Side-channel attacks
  - Lecture: Side-channel attacks on crypto implementations
  - Two Assignments
- Day 2: Advanced attacks
  - Lecture: Leakage evaluation: TVLA and alternatives + Assignment
  - Lecture: Profiling attacks + Assignment
  - Lecture: AI basics
- Day 3: AI and SCA
  - Lecture: AI-assisted SCA
  - Assignments
  - Lecture: Leakage simulators
  - Lecture: Higher order attacks (if there is time)
  - Unfinished Excercises

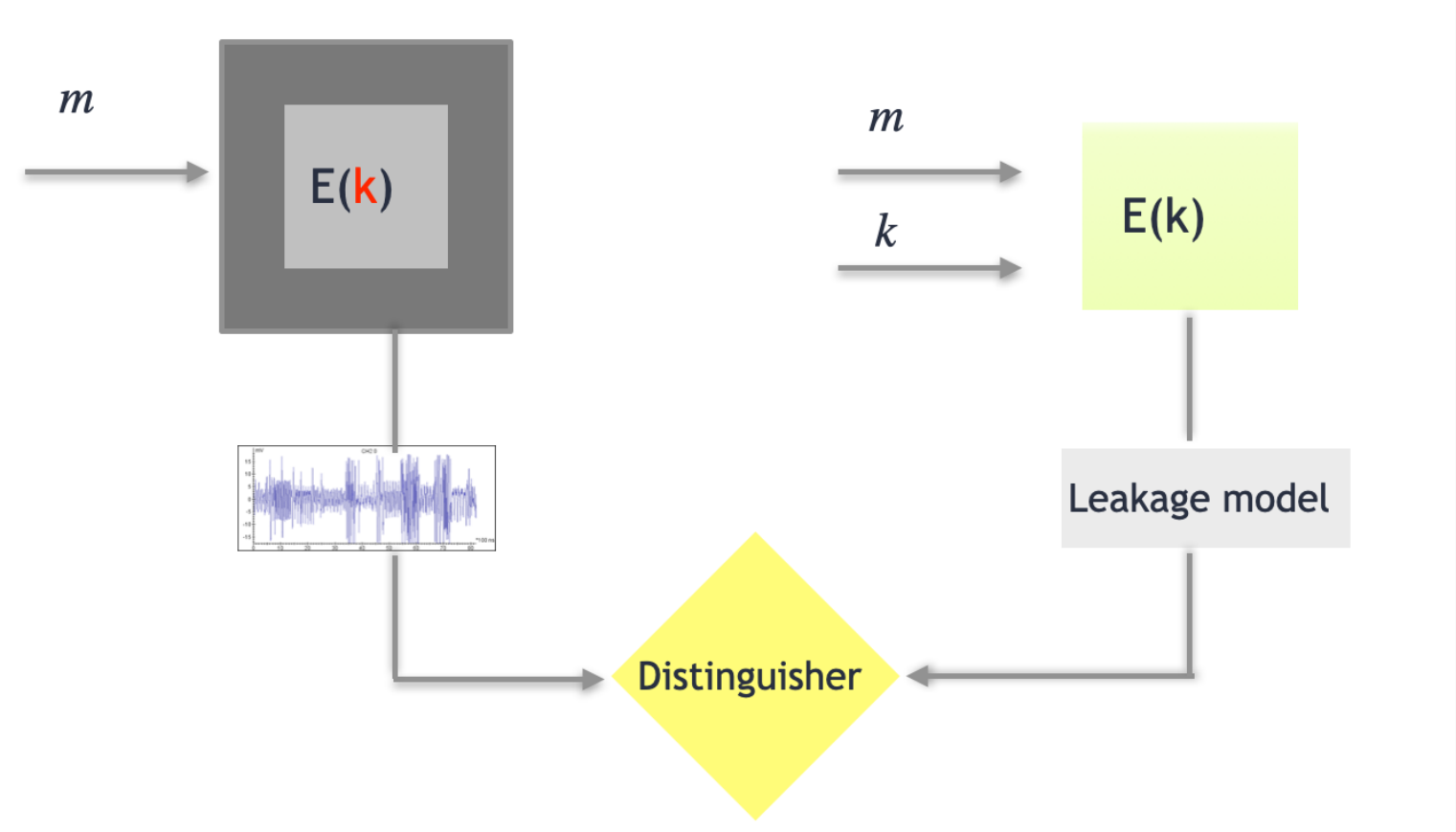
Day 2

- Recap and questions
- Lecture: Leakage evaluation: TVLA and alternatives
  - Assignment 1, TVLA: TVLA evaluation of (un)protected implementations
- Lecture: Template attacks
  - Step-by-step guide to profile and exploit a device's side-channel leakage
  - Assignment 3, Template Attack: Template attack on an ECC impl. on ARM Cortex microcontroller
- Lecture: AI basics
- Assignment 3, Deep Learning Attack: DL on an ECC impl. on ARM Cortex microcontroller
- Resources: `https://tinyurl.com/ysnznaka` [1]

[1] Full: https://www.dropbox.com/scl/fo/bd9r3lvzbk7eilqgytroo/ADK3b5aUicY2hzshgYCCQ8g?rlkey=5uzcphdvbnhlhs4z6zal63tqc&st=j5al094h

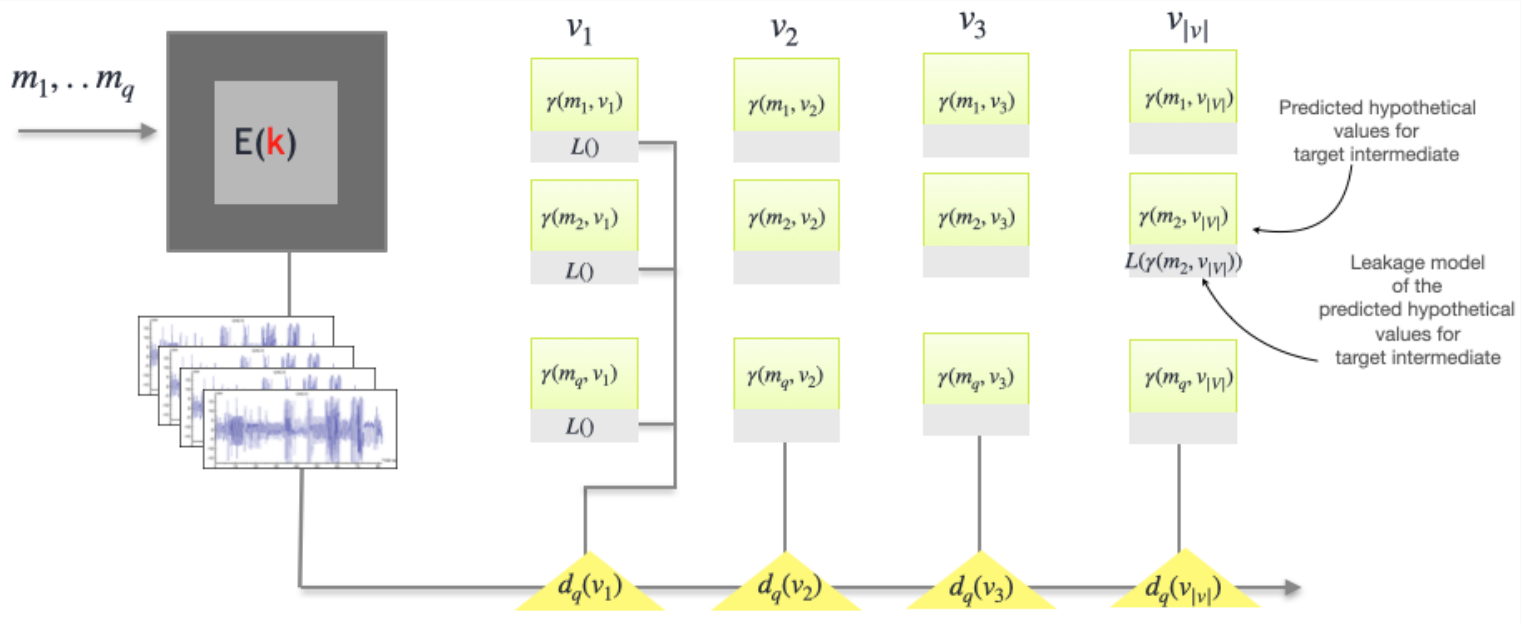
Recap

DPA attacks



- Can you name three types of leakage model functions?
- Can you name two distinguishers?
- What is missing in this figure?

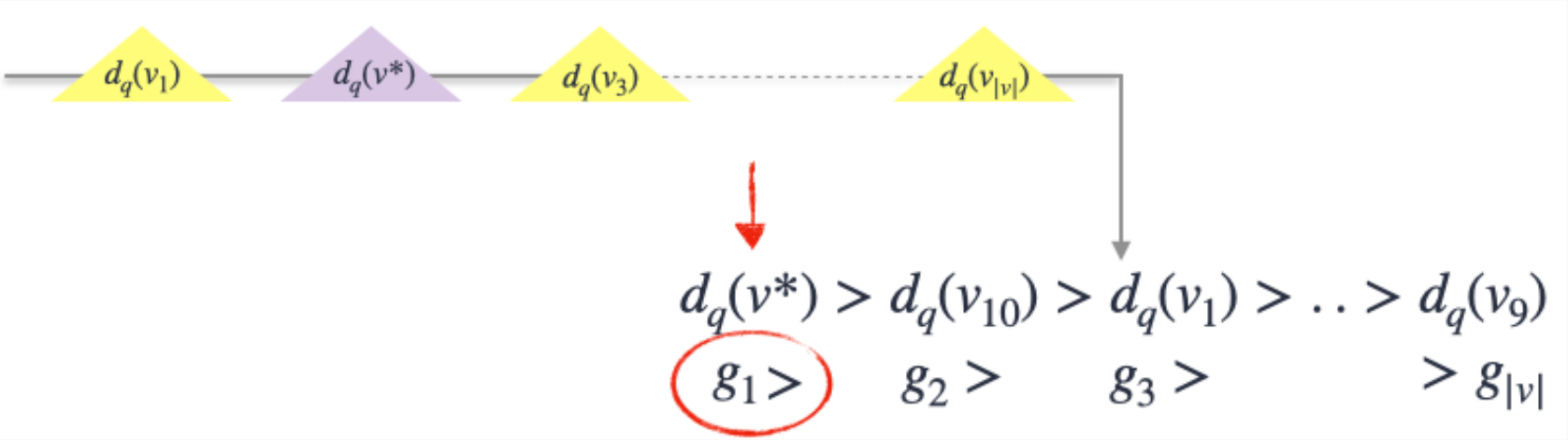
DPA attacks



- Where $g_i = d_q(k_i)$ is the score given by distingusher d to candidate v_i when q traces are used.
- The output of the distinguisher is the *guess vector* $g_q = [g_1, g_2, ...g_{|v|}]$.

DPA Success Metrics

Lets assume we have the results of a key recovery experiment with q queries. We know that the correct value is v^*:



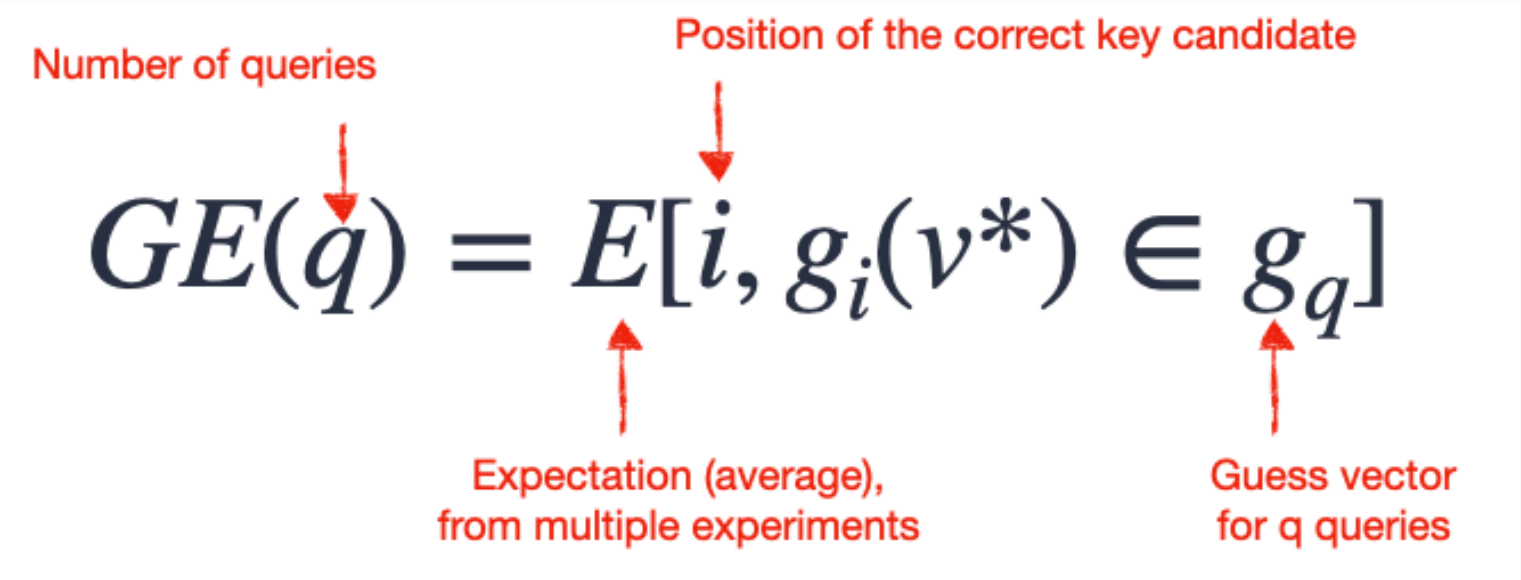
The result is the guess vector:

Position of the correct key candidate = 1

$$g_q = [g_1, g_2, g_3, \cdots g_{|v|}]$$

1. Guessing entropy/Key rank

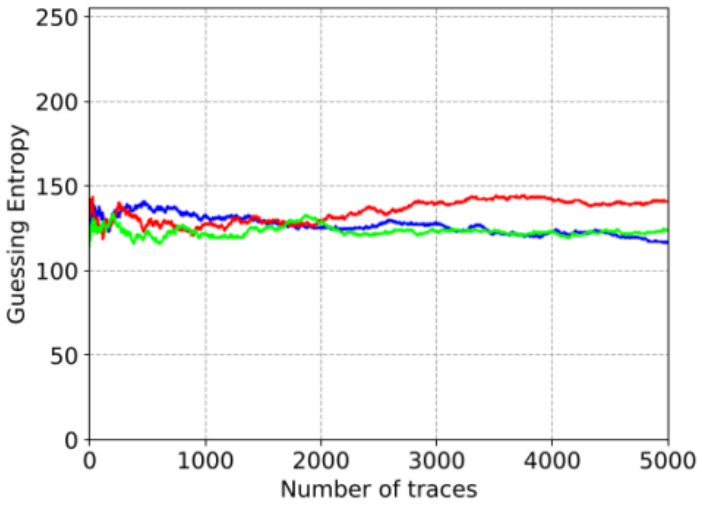
Guessing entropy gives the average position of the correct key candidate in a number of experiments. How do we compute it?



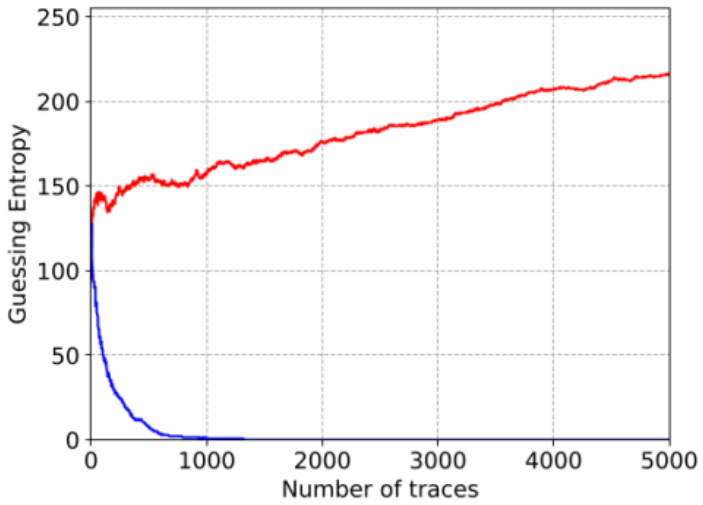
Why it is useful:

- Measures the *average* number of key candidates to be tested *after* a side-channel attack;
- Measures how much a side-channel attack reduces the complexity of an exhaustive key search.

Example 1: Guessing entropy in the wild



(a) CNN from [26] on the synchronized ASCAD dataset with random keys and the identity leakage model.



(b) CNN from [26] on the synchronized ASCAD dataset with a fixed key and the Hamming weight leakage model.

Fig. 2: Examples of GE behaviors with CNNs. The architecture and hyperparameters are not changed from the original paper except for the leakage model in Figure 2b.

Source for the figure: Lichao Wu and Léo Weissbart and Marina Krček and Huimin Li and Guilherme Perin and Lejla Batina and Stjepan Picek, *On the Attack Evaluation and the Generalization Ability in Profiling Side-channel Analysis*, 2020;

Example 2: Guessing entropy in the wild

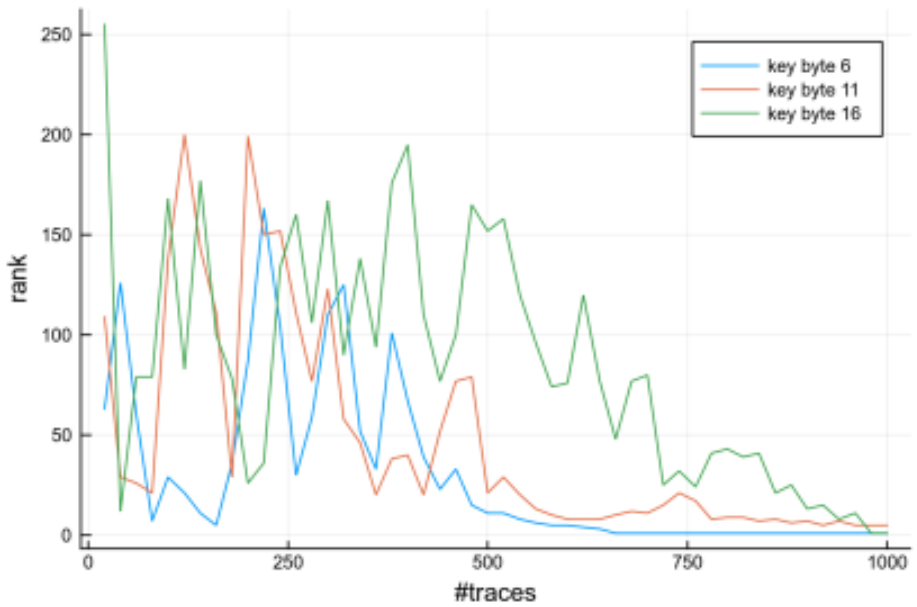


Figure 8: Key rank evolution for hardware AES engine FCA attack.

Source for the figure: Albert Spruyt, Alyssa Milburn, Łukasz Chmielewski, *Fault Injection as an Oscilloscope: Fault Correlation Analysis*, CHES 2020;

Leakage assessment

Leakage detection

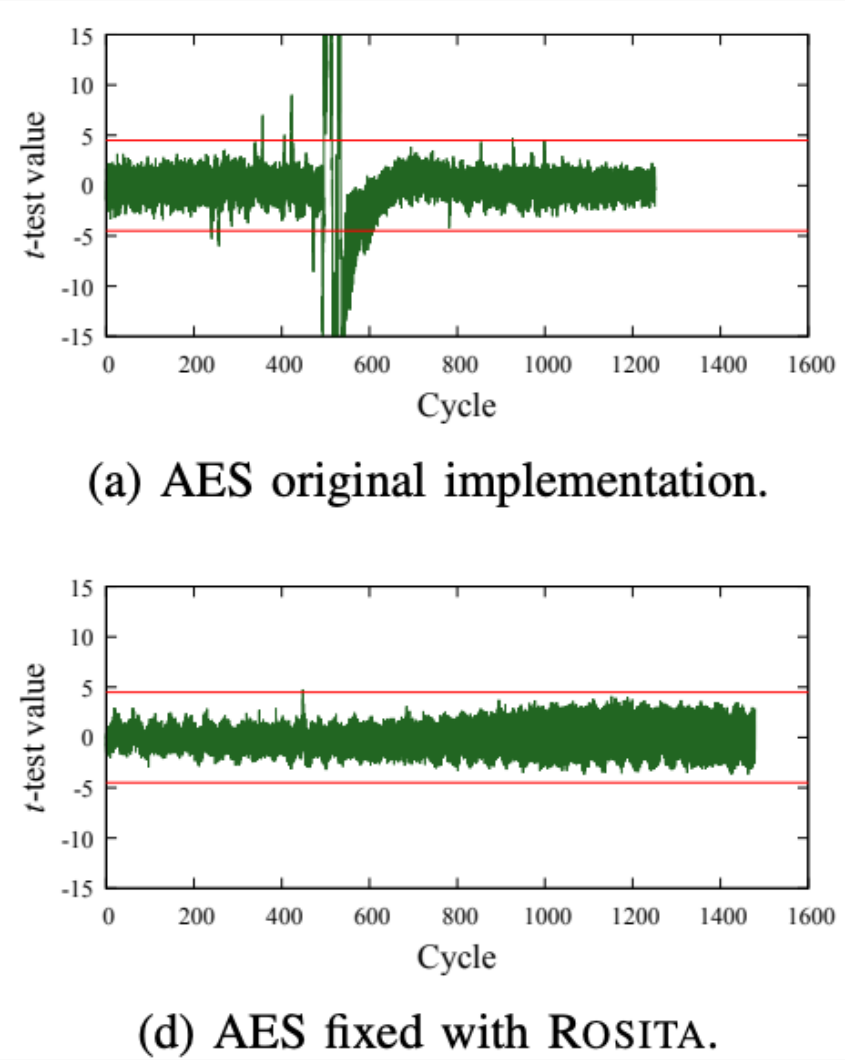
- You are a developer who wants to ensure that your implementation does not leak. Ideas?
  - Hint 1: we know that any dependency between the measured side-channel and the sensitive data is a potential vulnerability;
  - Hint 2: use reverse logic, if there are no dependencies, there is no side-channel vulnerability;
- Can we check vulnerability to side-channels without doing an attack?
  - yes! measure the side channel for different input values and see if they are different;
  - complicating fact: side channel measurements are influenced by many factors, not always straightforward;

TVLA - definitions

Test Vector Leakage Detection (TVLA) most popular leakage detection test.

- Non-specific or general test: aims to detect any leakage that depends on input data (or key);
  - a.k.a fixed - vs - random;
  - the topic of this lecture
- Specific test: targets a specific intermediate value of the cryptographic algorithm that could be exploited to recover keys or other sensitive information.
  - a.k.a fixed - vs - fixed;

Example: TVLA in action



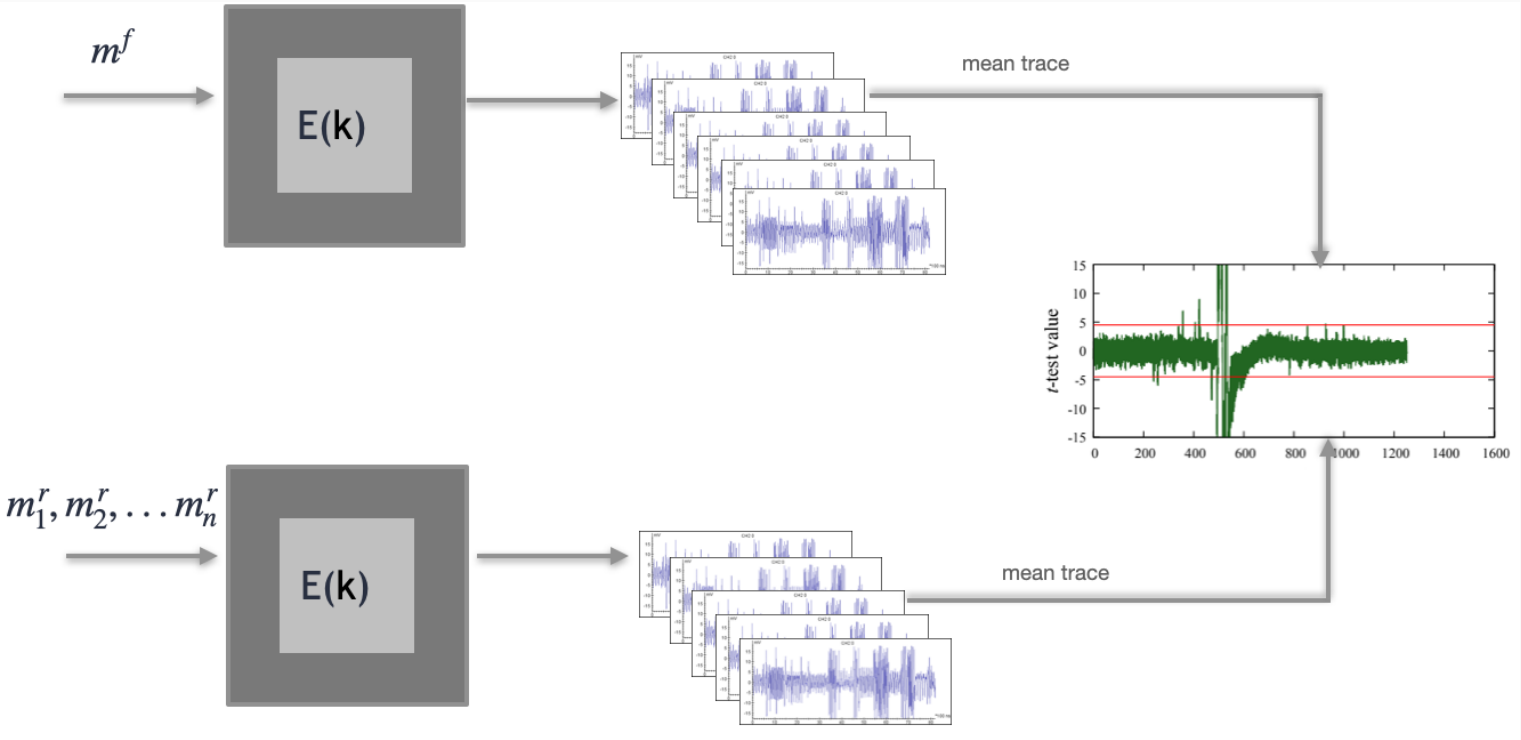
(a) AES original implementation.

(d) AES fixed with ROSITA.

Source for the figure: Madura A Shelton and Niels Samwel and Lejla Batina and Francesco Regazzoni and Markus Wagner and Yuval Yarom *Rosita: Towards Automatic Elimination of Power-Analysis Leakage in Ciphers*, NDSS 2021

Step 1: data preparation

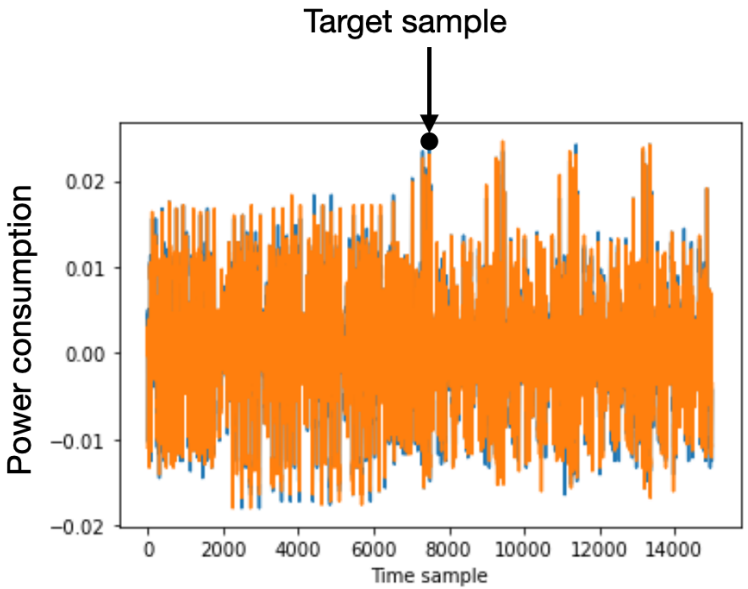
Collecting traces for TVLA test:

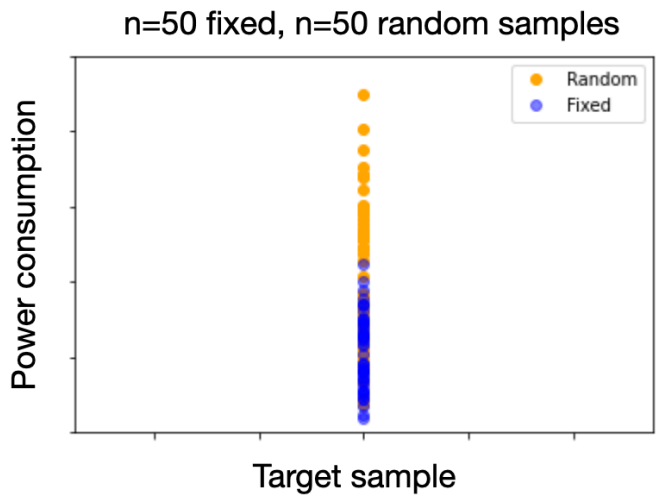


IMPORTANT: This version of TVLA is called fixed - vs - random, and is a *two-sample t-test*.

Step 2: computing the t-score for the target sample

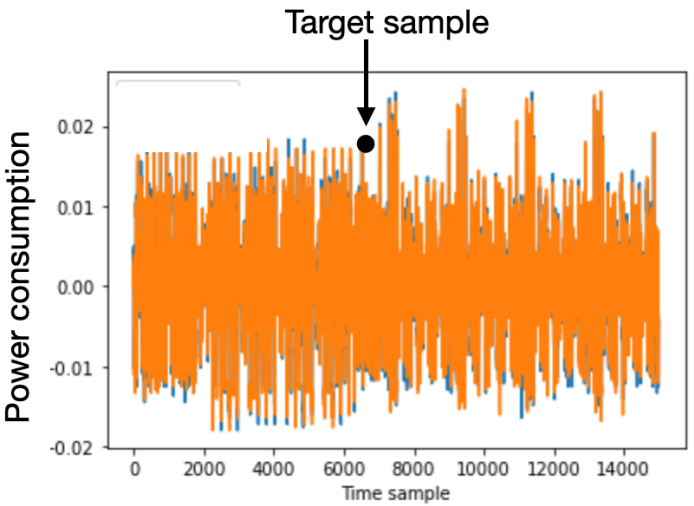
Using simulation, we create two sets of data : set f corresponding to the *fixed input*, and set r corresponding to the *random input*. Each set has a *sample size* of $n = 50$ observations.

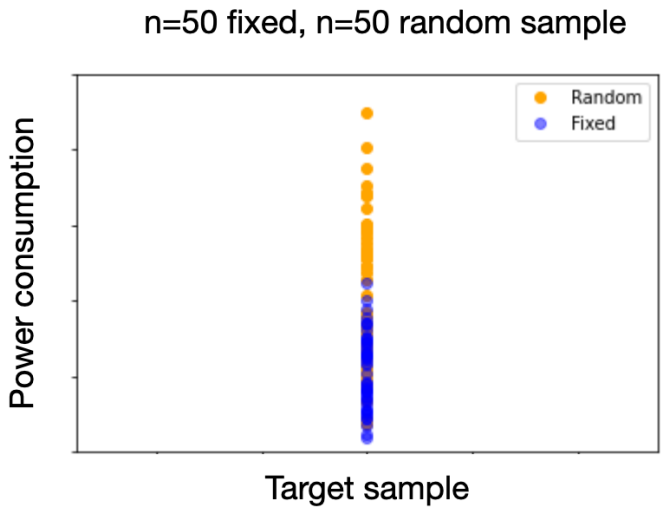


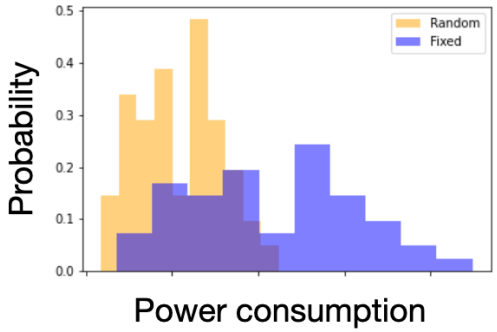


Step 2: computing the t-score for the target sample

Using simulation, we create two sets of data : set f corresponding to the *fixed input* and set r corresponding to the *random input*. Each set has a *sample size* of $n = 50$ observations.



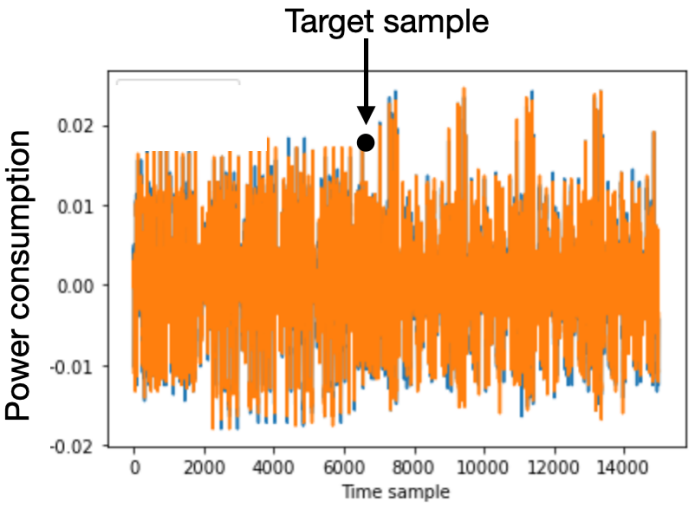


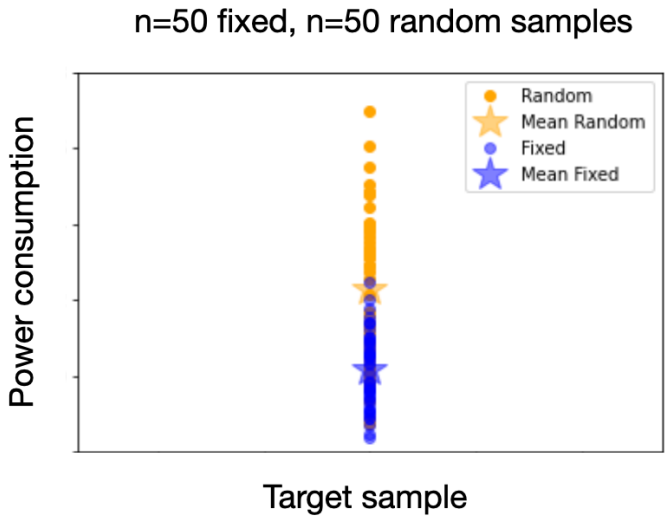


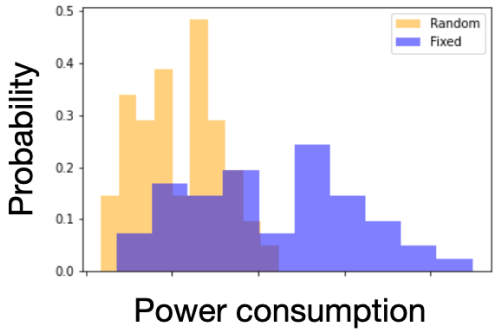
Step 2: computing the t-score for the target sample

For each set, we compute the mean values:

- $\overline{x}_f$ is the mean for the set f, computed as $\overline{x}_f = \frac{1}{n}\sum_{i=0}^{n} x_i$, where $x_i \in f$;
- $\overline{x}_r$ is the mean for the set r;



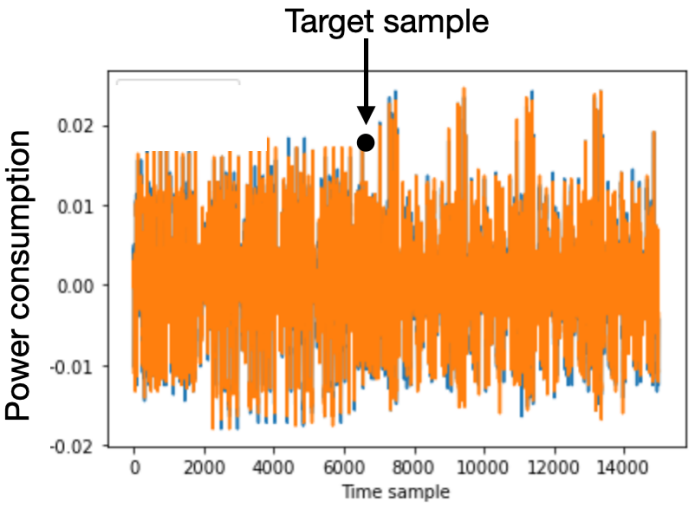


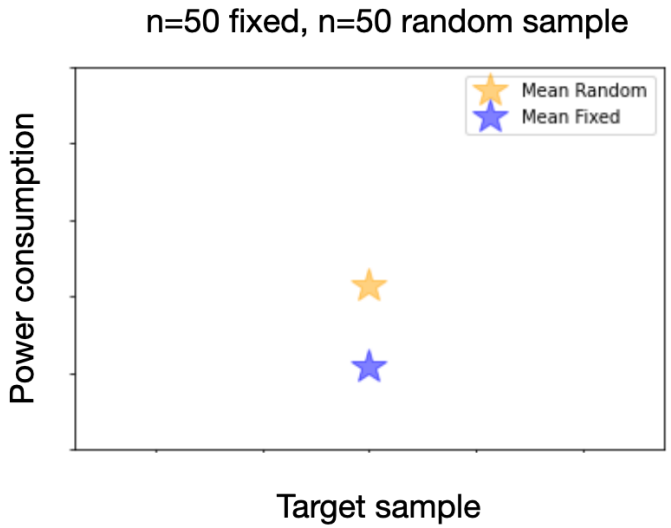


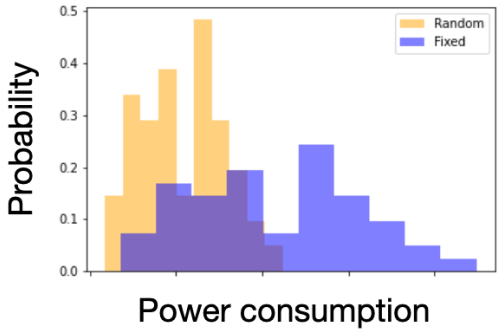
Step 2: computing the t-score for the target sample

For each set, we compute the mean values:

- $\overline{x}_f$ is the mean for the set f, computed as $\overline{x}_f = \frac{1}{n}\sum_{i=0}^{n} x_i$, where $x_i \in f$;
- $\overline{x}_r$ is the mean for the set r;

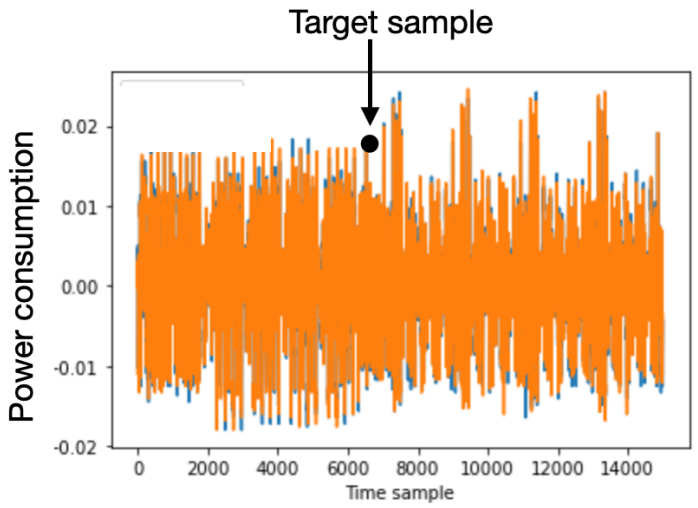


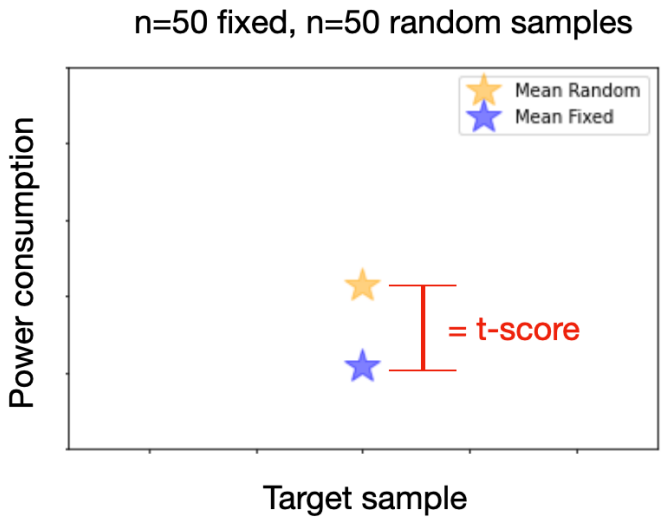


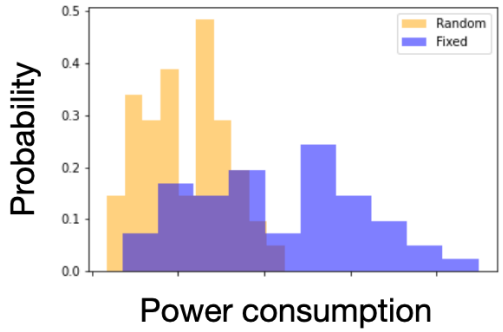


Step 2: computing the t-score for the target sample

The *t-score* is the standardized difference between the two mean values $\overline{x}_f$ and $\overline{x}_r$.



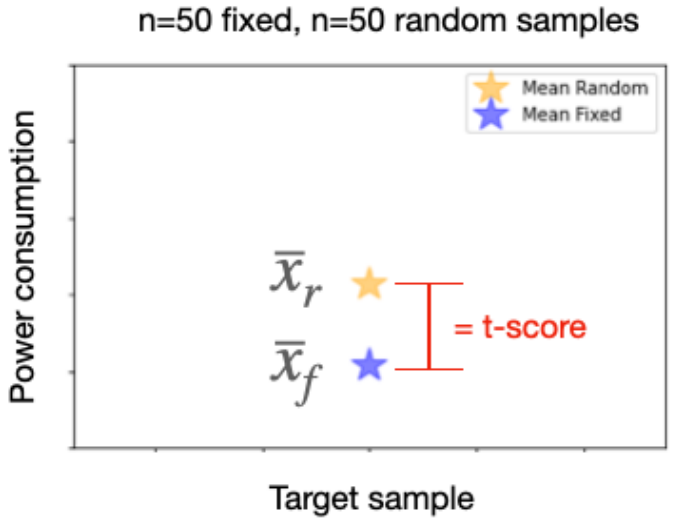




Step 2: computing the t-score for the target sample

What is a standardized difference?

$$t = \frac{\bar{x}_f - \bar{x}_r}{\sqrt{\left(\frac{s_f^2 + s_r^2}{n-1}\right)}}$$



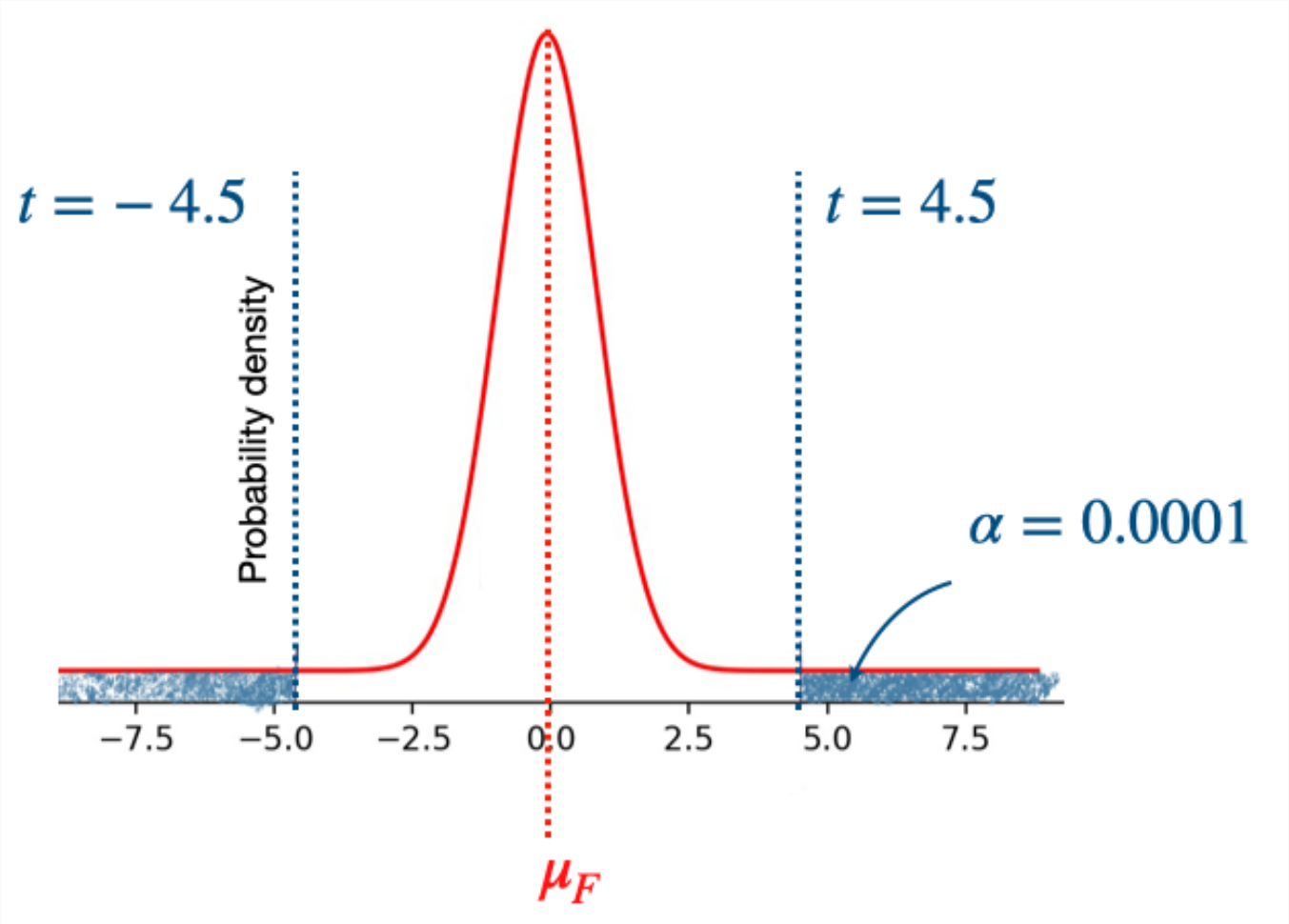
$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad \bar{x} = \frac{1}{n}\sum_{i=0}^{n} x_i$$

This version of the test is called *Welch's* t-test and is applied for equal or unequal sizes for the fixed and random sets and $s_f \neq s_r$.

Step 3: repeat Step 2 for all samples in the trace.

Done!

Question 2: Why the 4.5 threshold?



- If H_0 is true, the probability that the t-score is in the acceptance region is 0.999
- The probability that the device is not guilty, but the evidence shows otherwise is very small (99.99%)

Q3: is TVLA the only leakage detection method?

- No, many more tests available:
  - χ-square, used for categorical data;
  - F-test, when we test multiple means;
  - Mutual information, tests (in)dependence between two variables;
  - Correlation coefficient, reduces the risk of false negatives in fixed-versus-random t-test;
  - Deep learning
  - etc..
- To choose wisely, we need to know what are the assumptions, the use case and understand H_0.

Further reading

- François-Xavier Standaert, Tal G. Malkin, Moti Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, EUROCRYPT, 2009; introduces the formal definition for success rate of order o and guessing entropy for side channel attacks;
- Stefan Mangard, *Hardware Countermeasures against DPA - A statistical Analysis of Their Effectiveness,* CT-RSA 2004, introduces the concept of Signal-to-Noise Ratio (SNR) to the area of side-channel analysis;
- Stefan Mangard, Elisabeth Oswald and Thomas Popp, *Power Analysis Attacks, Revealing the Secrets of Smart Cards (Chapter 4)*,2007 provides an in-depth treatment of deriving SNR for different use cases;
- G. Goodwill, B. Jun, J. Jaffe and P. Rohatgi, *A testing methodology for side-channel resistance validation,* CRI, 2011, `https://www.rambus.com/papers/security-papers/dpa-countermeasures-security-papers/` is the official documentation for the TVLA framework;
- Carolyn Whitnall, Elisabeth Oswald, *A Cautionary Note Regarding the Usage of Leakage Detection Tests in Security Evaluation*, 2019, `https://eprint.iacr.org/2019/703`, provides an in-depth analysis on the limitation of TVLA.

Do you want to get a better understanding?

- Check: "Hypothesis testing for leakage assessment in side channel analysis" by Ileana Buhan
  - https://cescalab.cs.ru.nl/hypothesis-testing-for-leakage-assessment-in-side-channel-analysis/