Applied Security

Leakage Detection

These slides were prepared as part of the REASSURE tutorial on leakage detection (given at CARDIS 2018) by Carolyn Whitnall, Valentina Banciu, and myself.

What is ‘leakage detection’ as opposed to ‘leakage attack’

DPA attacks exploit the fact that different inputs to the cryptographic system produce different measurable leakages.

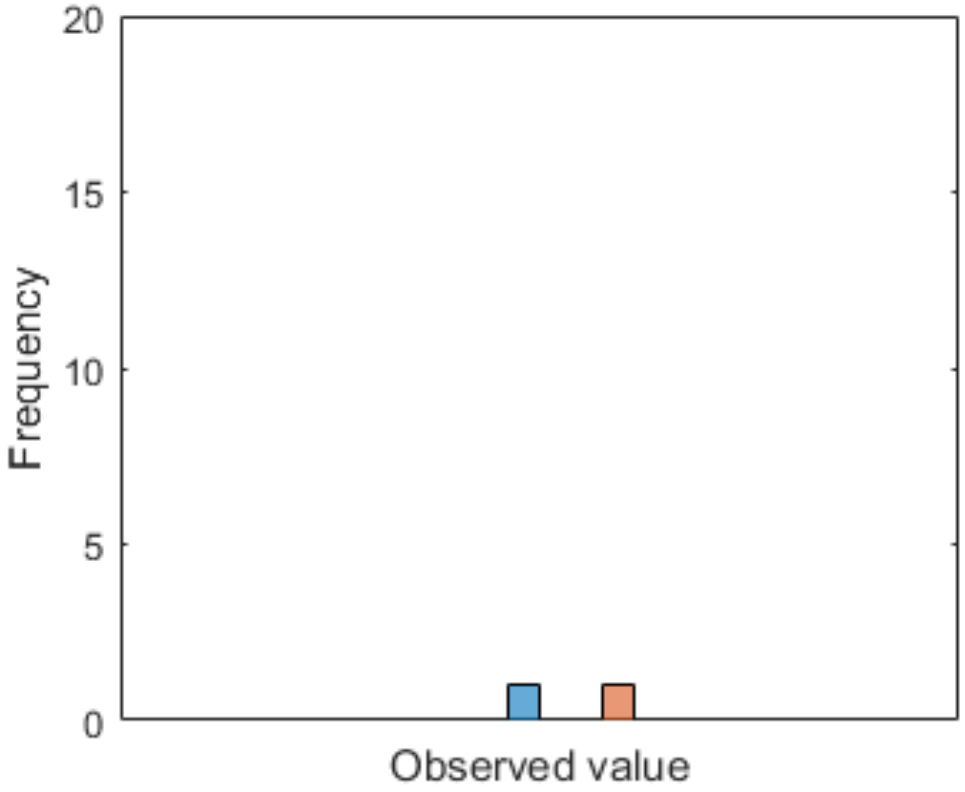
- Leakage detection is about checking whether or not this is the case.
- Leakage attacks are about demonstrating that leakage is exploitable and leads to a concrete vulnerability.

What is ‘leakage detection’ as opposed to ‘leakage attack’

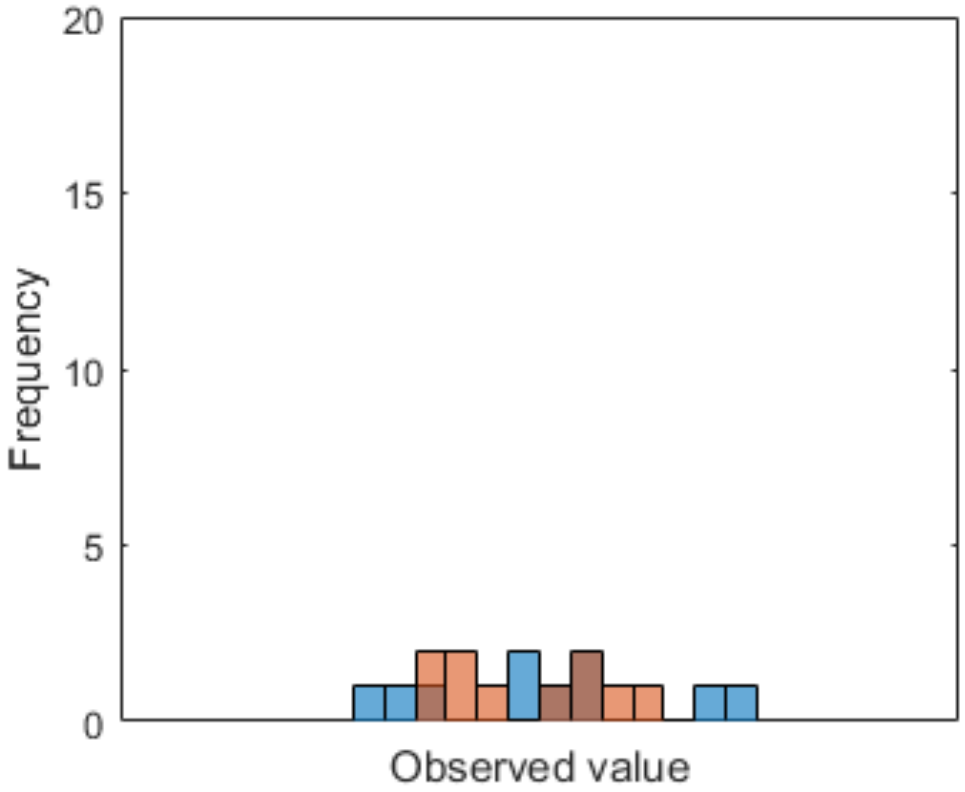
DPA attacks exploit the fact that different inputs to the cryptographic system produce different measurable leakages:

- If different inputs to the system always trigger (on average) the same power consumption then there is nothing for an attacker to exploit.
- Thus: measure the power consumption for different inputs, and see if it is different.
- Unfortunately, it is made complicated by the fact that the power consumption is a noisy process, influenced by many factors, of which the inputs may or may not be one.

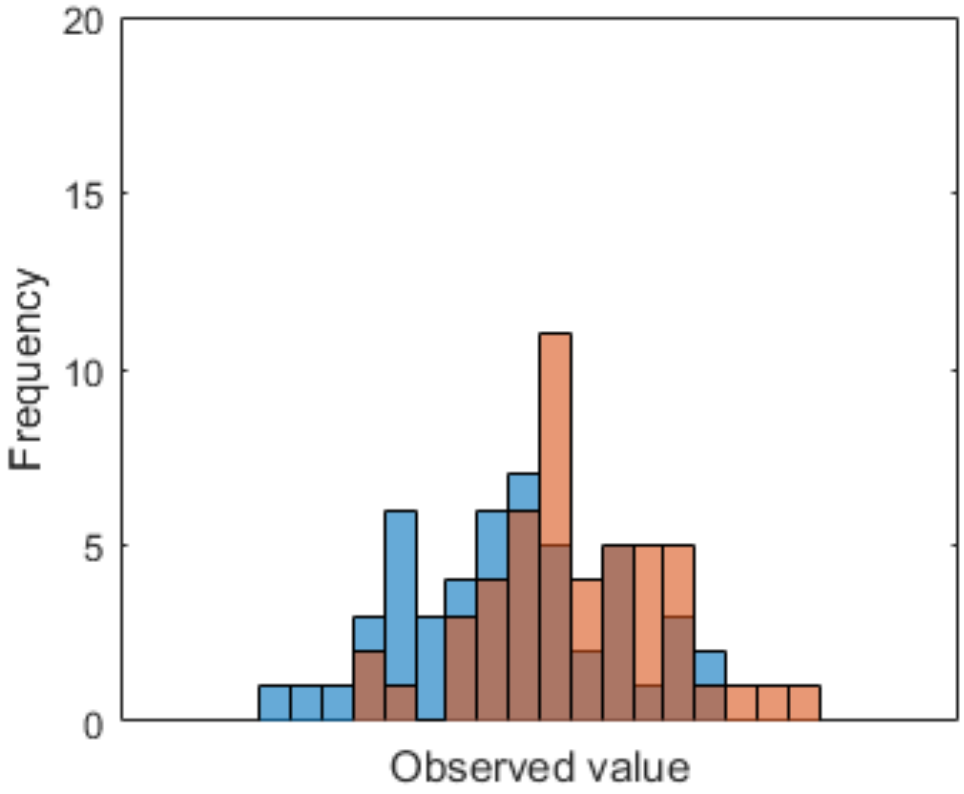
Experiment (1/6): Suppose we have two observations. How can we tell if they are from the same underlying random process, or from different processes?



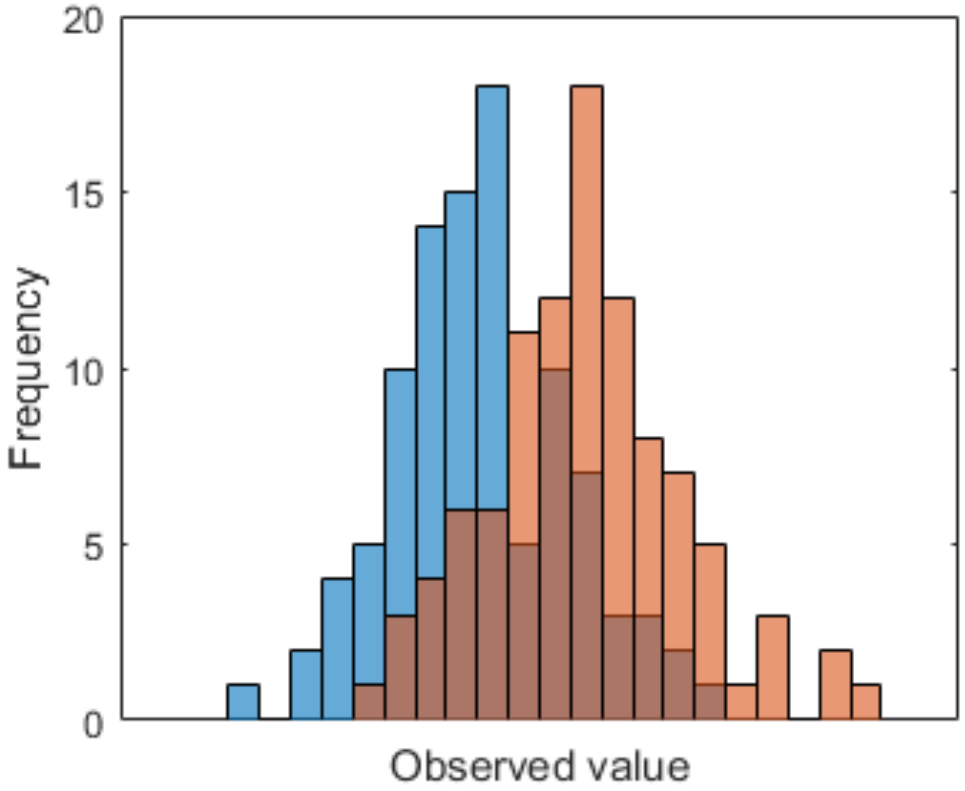
Experiment (2/6): The answer is, we can't! Not definitively. The values are different but if we collect a few more (say, 10) observations of each we find that they are all mixed up together with no clear pattern.



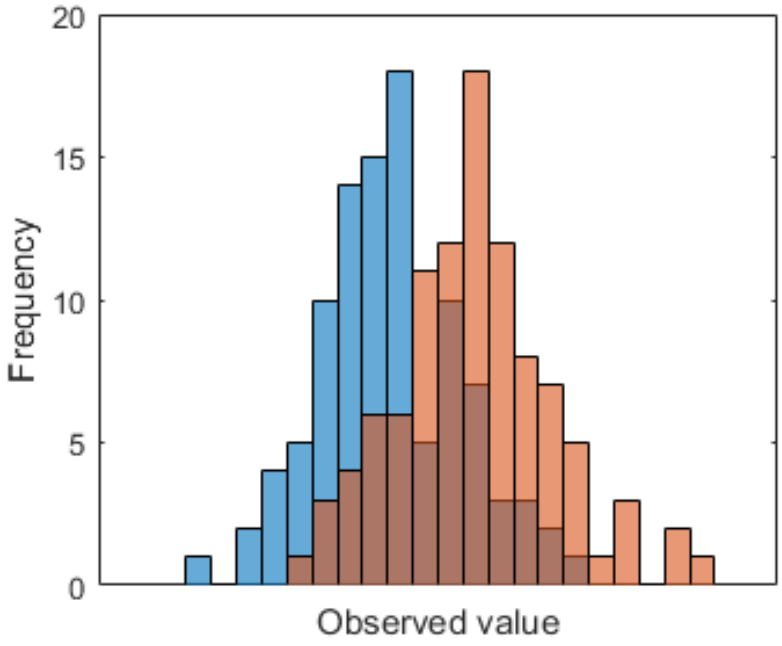
Experiment (3/6): Observe 50 of each and we start to see, perhaps, some more discernible tendencies emerging…

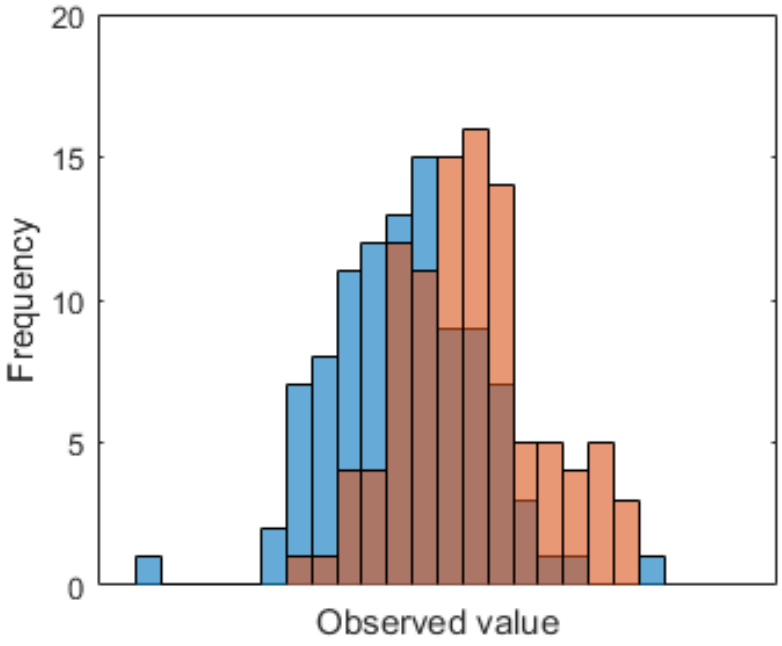


Experiment (4/6): With 100 observations from each, we might suddenly feel a lot more confident that the orange and the blue values are from different processes, with different means

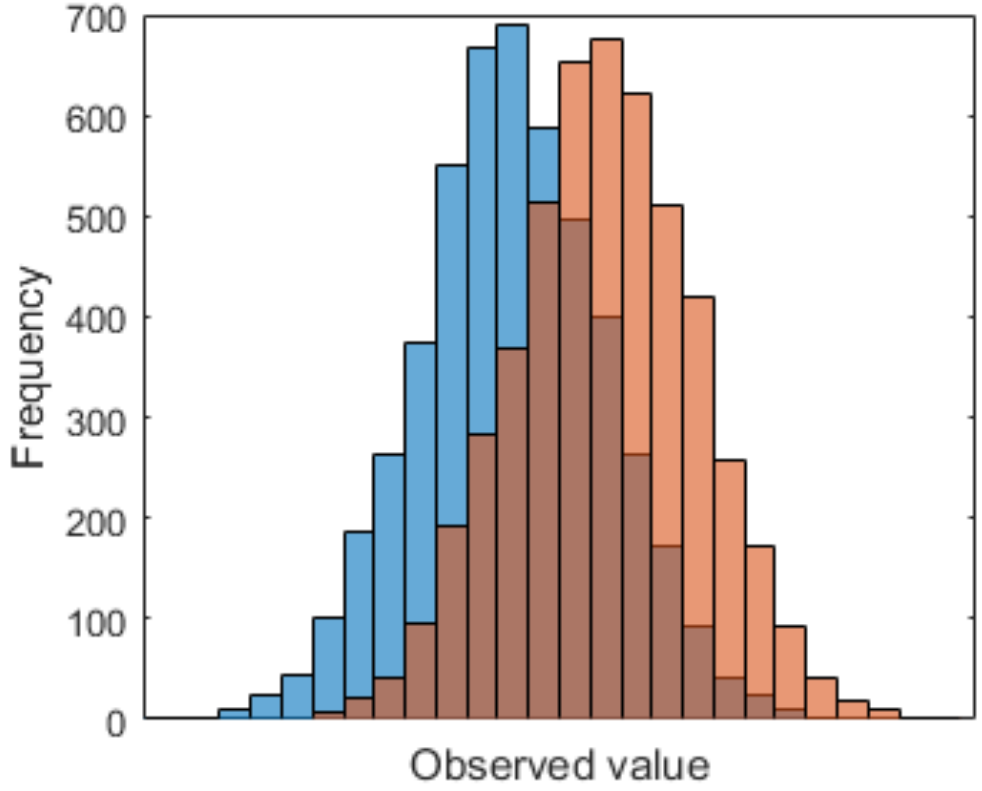


Experiment (5/6): But what if the figure on the right also depicted two samples of 100 observations, generated in exactly the same way? Do we still feel as confident concluding that there are two distinct processes?





Experiment (6/6): With even more data (5,000 observations here) the uncertainty decreases. Indeed, the processes are different in this case, and because as it happens they are both normal, the familiar bell curves emerge.



Using the sample means to decide

- What thought processes are you using to decide whether the two are the same or different?
  - Instinctively, we look at how far apart the middles of the samples are, and how spread out the rest of the observations are.
- Let's suppose from now on that all the processes in question are normally distributed. This means that the middle corresponds to the mean, and also gives us some nice statistical properties that will help us as we go.
- The means might look well separated, but they are only estimates. The more spread out the observations are, the greater the danger that those estimates are imprecise (i.e., the true means might be the same after all).
- So what we really want to know is: *are the estimates of the means precise enough so that the distance between them can be taken as proof that the true means are truly different?*

The estimated means themselves vary!

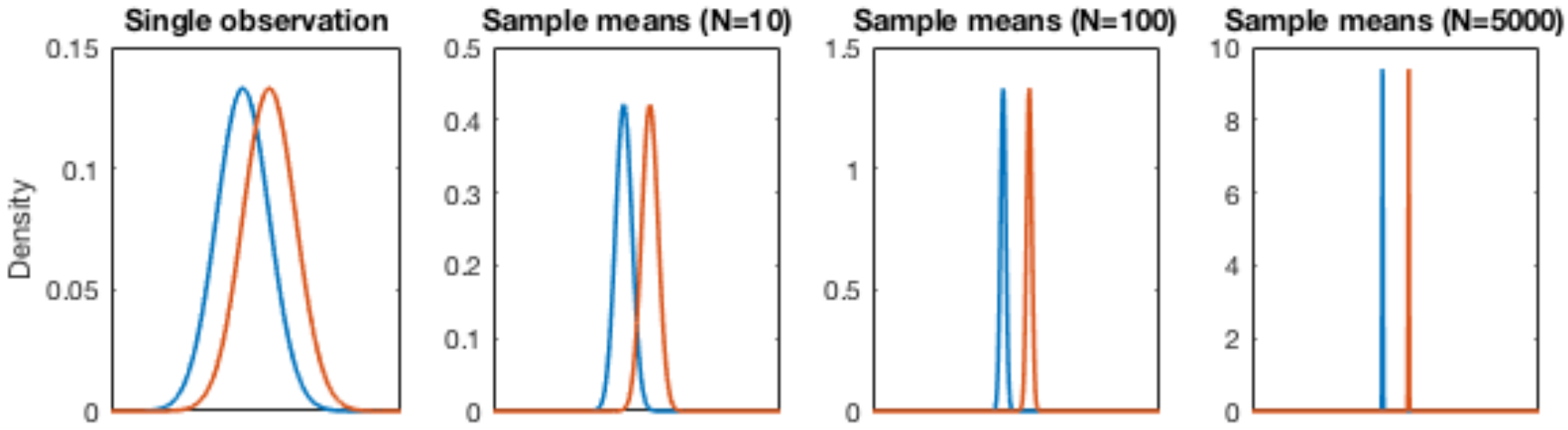
To help us answer this, we need to understand that the estimated mean itself has a distribution.

- We encountered this concept before and referred to it as the sampling distribution

If we were to repeatedly draw the sample and estimate the mean, it would range either side of the true mean to some extent.

- The larger the true variance, the more the estimate of the mean will vary.
- As the sample size increases, the estimates will become more precise (that is, more tightly clustered around the true mean).

For example, the distributions of the sample means corresponding to the growing numbers of observations we looked at earlier can be drawn as follows:



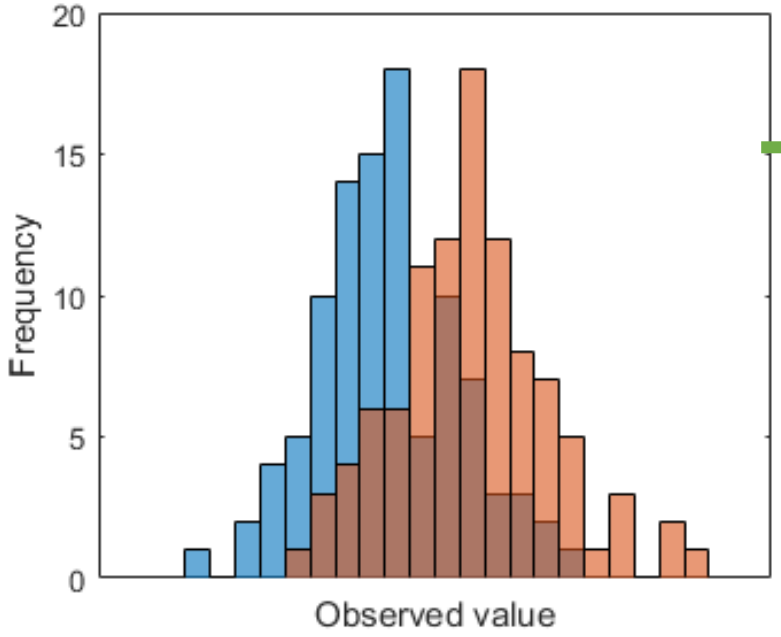
In practice, we only have one sample, so we don't actually know what the sample mean distribution looks like! But we can estimate it with the help of the sample variance…

How far can the sample mean vary from the true mean?

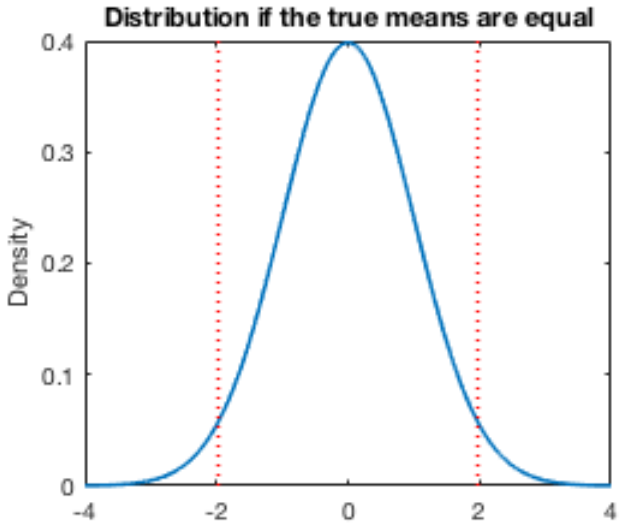
- The sampling distribution can tell us:
  - Our estimate for the standard deviation of the sample mean is $s/\sqrt{N}$, where *s* is the sample standard deviation and *N* is the sample size.
- The question of whether two samples are from the same or different processes can thus be thought of in terms of the **overlap** between the two estimated distributions of the sample means: *is it small enough to conclude that the true means are truly different?*

The *t*-test

- The *t*-test provides formal decision criteria for whether or not the sample means are different enough to suggest a difference in the true means.
- Actually, the question the *t*-test asks is *whether or not the (standardised) difference between the means is big enough to suggest that the true difference is non-zero.*
  - *We have asked this question before in the context of determining an estimate for the number of needed traces to succeed with DPA style attacks*
- This standardised difference also has a distribution of its own. Without going into too much this distribution can be used to provide a threshold value.
  - If the observed difference is bigger than the **threshold** value, the real difference is non-zero – subsequently, the two samples are generated by different processes.



$$t = \frac{mean(orange) - mean(blue)}{\sqrt{\frac{var(orange)}{N_{orange}} + \frac{var(blue)}{N_{blue}}}}$$



The *t*-statistic is computed from the samples and compared with the appropriate *t*-distribution. If it is larger in size than some chosen lower bound for unlikely values then the test concludes that the true means are different.

Using the *t*-test to detect leakage

- The *t*-test (an ditto a correlation test) can be used to decide if two samples were generated by underlying processes with different means.
- In general terms, the strategy for leakage detection via the *t*-test is to feed different data inputs into the device, and then check to see if the corresponding side-channel measurements are detectably different.
  - There is no single definitive way to do this. In particular, the *t*-test compares precisely two samples and there are any number of data-dependent ways to partition up the measurements in order to make the comparisons.
  - A good strategy will be one that is **comprehensive** (i.e. covers all possible vulnerabilities) and **achievable** (i.e. within data and computation budgets).

Test Vector Leakage Assessment (TVLA)

- Goodwill et al.'s Test Vector Leakage Assessment framework [GJJR11] is a set of recommendations aiming to achieve a good strategy.
- Tests fall into two categories:
  - **Specific tests** target known intermediate values. Traces are collected for uniformly random inputs and partitioned according to bits or bytes of the data state. To compare the partitions, *t*-tests are performed. A limitation is that such an approach is only sensitive to what is targeted.
  - **Non-specific tests** aim to catch more general vulnerabilities. The fixed-versus-random method tests for differences between traces collected for random inputs and traces with a fixed input. A limitation is that the precise source of the vulnerability is not always clear, making it difficult for an attack to exploit or a designer to fix.

[GJJR11] G. Goodwill, B. Jun, J. Jaffe and P. Rohatgi, *A testing methodology for side-channel resistance validation*, CRI, 2011.

TVLA recommendations

The TVLA framework specifies a threshold of **4.5** for deciding that a particular point in the trace is leaky. It is outside of the scope of this unit to interrogate this choice:

- It could guard against false positives in the context of multiple comparisons, but does so at the expense of guarding against false negatives
- According to TVLA, all detected vulnerabilities should be confirmed by repeating the set of tests on an independently-generated second acquisition. Again this may guard against false positives

TVLA recommendations

The framework also specifies some requirements regarding data acquisition and trace processing.

It relies on known facts about the t-test, but ignores that the use case here is in the context of a multiple comparison problem with unknown dependencies.

Thus the test is as such not suitable for definitive conclusions, but it is rather useful for some simple testing.

The TVLA fixed-versus-random

- Suppose a device leaks values proportional to the Hamming weight of the intermediate values, plus some Gaussian noise.
- Then the average leakage value as the inputs vary uniformly at random will be 4.
- Meanwhile, if an input is chosen such that the target intermediate value has a Hamming weight of 7, the average leakage value for repeat executions with this input will be 7.
- A *t*-test with large enough sample sizes of each (relative to the noise magnitude) should be able to conclude that there is a true difference, i.e. a leak.

Discussion

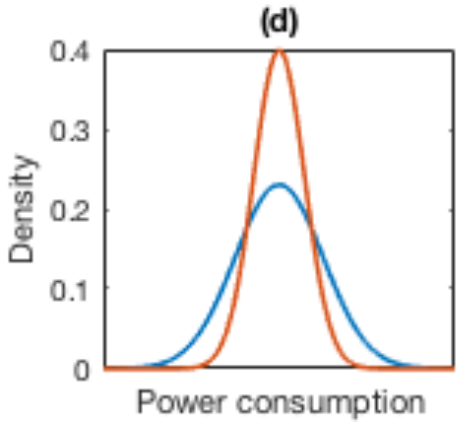
- Might a test fail?
- Might one test do better than another?
- Is noise bad news?

Discussion

- Might a test fail?
- Might one test do better than another?
- Is noise bad news?

Why a test might fail

- If the mean associated with the fixed input coincides with the overall mean as the plaintext varies, then although the distributions are different (a fixed input means a smaller variance) the *t*-test will not be able to distinguish between them as it is only sensitive to mean differences.



  - Consider, e.g., a device with power consumption proportional to the Hamming weight, in the event that the fixed input produces an intermediate with Hamming weight 4.

Nature of statistical tests

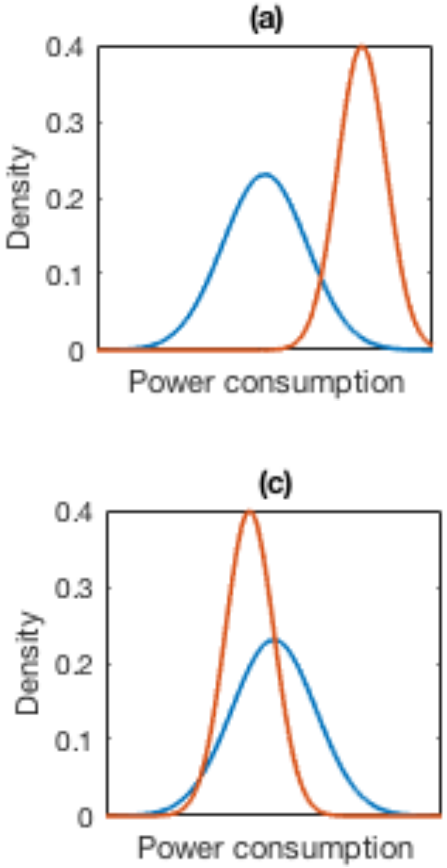
- Another reason a test can fail is that statistical tests are inherently probabilistic. The decision criteria are designed to minimise errors (and trade between them); it is not possible to eradicate them.
  - For example, there are test scenarios and configurations which will sometimes detect and sometimes fail to detect, even though the leakage is always there.
  - There will also always be a certain proportion of false positives, i.e. tests which conclude that there is leakage when there isn't.
- Controlling false positives and false negatives is important but TVLA fails to provide guidance.

Discussion

- Might a test fail?
- Might one test do better than another?
- Is noise bad news?

Why one test might do better than another

- The choice of fixed inputs in examples (a) and (c) both produce means that are distinct from the overall mean as inputs vary. However, the *distance* is much greater for (a) than it is for (c).
  - For example, a fixed input that leads to a Hamming weight of 8 implies four times the distance from the overall mean as one that leads to a Hamming weight of 3.
- The numerator of the *t*-statistic will be smaller in scenario (c), which implies that we will need larger samples than in scenario (a) before we will be convinced that there is a real difference.
- You can also think of this as needing to squeeze the sampling distributions more before they cease to overlap



Discussion

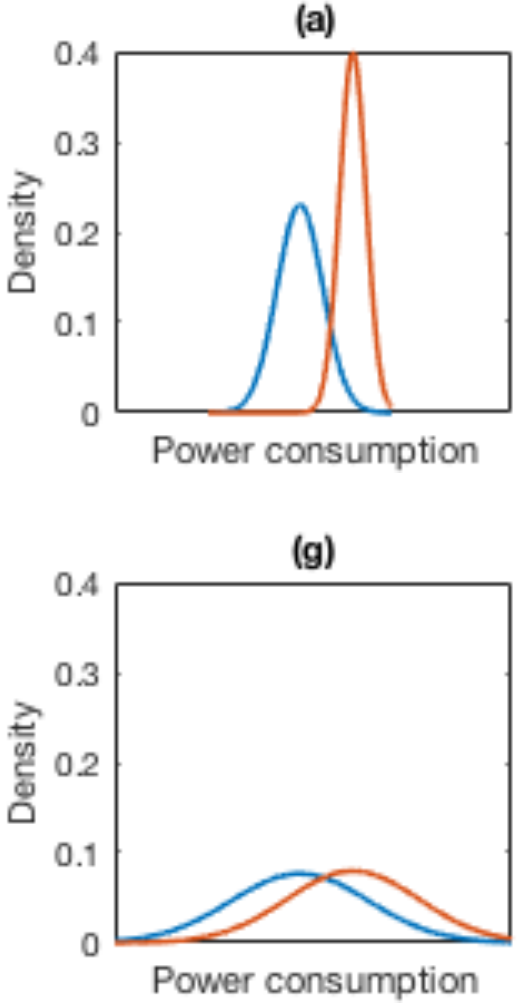
- Might a test fail?
- Might one test do better than another?
- Is noise bad news?

Why noise is bad news

- Remember from previous lectures that the data inputs are only one potential source of variation in the power consumption.

$$P_{total} = P_{op} + P_{data} + P_{noise}$$

- The more noise there is from other processes on the device, and from your measurement set-up, the greater the overlap between the distributions, even for the same choice of fixed input.
- A different choice of fixed input might sometimes help to some extent, but larger noise generally means that more data is needed before the sampling distributions of the mean are adequately separated.



Summary

- The *t*-test is effective at detecting certain forms of data-dependency in side-channel measurements.
- We have touched on the fact that the number of traces has a bearing on the outcome of a test, and that lots more traces are needed if the signal is small or the noise large (or both).
- TVLA provides a useful (but incomplete) set of recommendations for practical leakage detection.
- The fixed-vs-random test is though a very useful tool for testing your (protected) implementation for leaks
  - If you compare your protected and unprotected implementation using fixed-vs-random, you should see a decrease in the number of leaks.