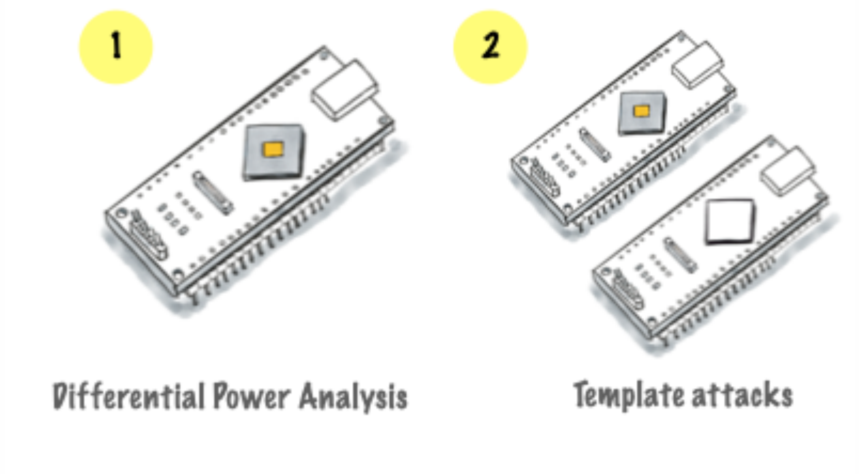
In this post, we explore the most powerful side-channel attacks, in other words, template attacks. We assume the reader to be familiar with how differential power analysis (DPA) attack work. This tutorial will be handy to readers who bravely attempted to tackle template attacks but were crushed by a heavy math description. In this story, the authors and the reader play the role of the adversary, who executes a template attack to recover an unknown key. So the pronoun "we" in this text will be used interchangeably with "the adversary."



In contrast to a DPA attack, where the adversary has one device with an unknown key, a template attack requires two *identical* devices. To recover the device's key programmed with the unknown key, the adversary can use a second device to learn from. It is assumed that we, the adversary, can program the device with any key we like.

Dating a Template

Template attacks follow the same principle as any respectable dating app. All dating apps have a database of users interested in finding a match. Each user needs to describe some attributes which aim to simultaneously make the user unique and put them in the best possible light. To make the analogy more intuitive, let us assume the dating app contains all available dating candidates when it launches. For pro users, the dating app has an additional feature. Using a predefined score, like music taste, shared hobbies, or anything else, the app can shortcut the search phase and simply list "best matches."

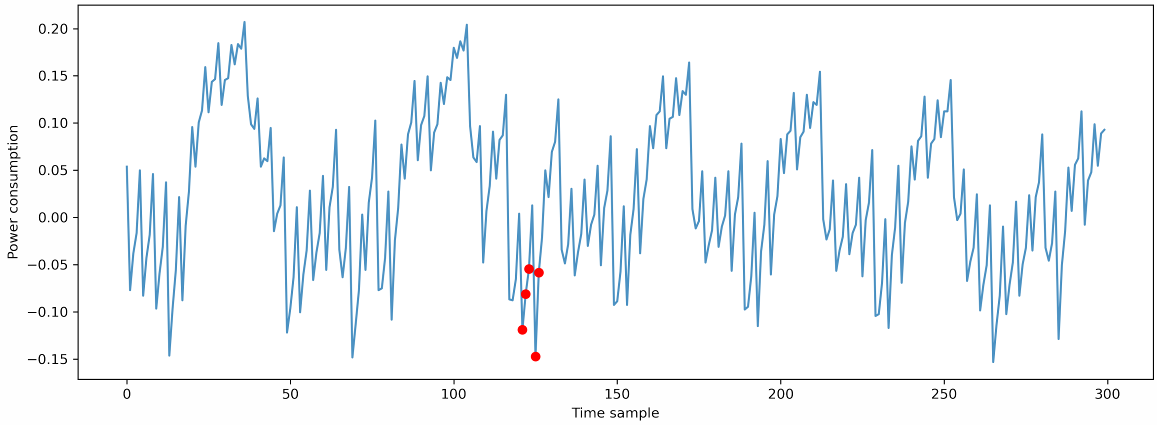
What to template?

As attackers our first job is to decide what to attack. We have two options, we can build template for pairs (p, k) where p stands for the plaintext and k is the key. Alternatively we can also target an intermediate value v (e.g. S-box out, round out, etc.), which can be used to determine the value of k, if we know p. This second option is preffered because it reduces the amount of candidates to describe. As keys for modern algorithms are typically very large, we break the problem into more manageble bites or ahem.. bytes. To retrieve v, we will target one byte at a time, which we refer to as the *byte of interest*. To recover all bytes of v we repeat the process for all bytes.

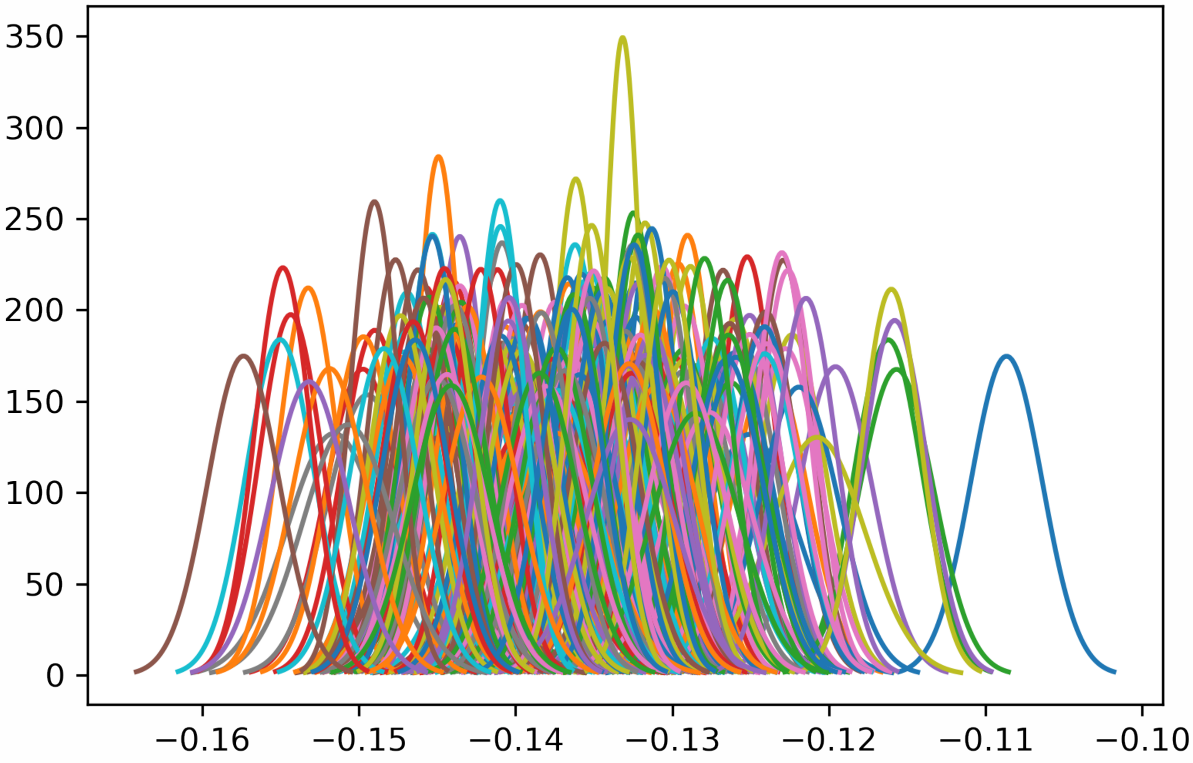
Template buiding

Like the dating app, a template attack needs a database of potential candidates. The construction of this database is known as the *template building phase* when the profile or *the template* of each candidate is created. A *template* is a description of some sort for a candidate. To create a template, we will program the device we control with all possible values for the byte of interest, and we will collect the side-channel traces. We use the collected side-channel paths to show and describe each candidate. You guessed correctly, we need 256 templates to recover one byte!

To distinguish between the different templates let us use h_i the template for value i. According to [xxx] the *points of interest* are those points in a side channel trace that contain most information about the *byte-of-interest*. While the number and choice of the points of interest is vital to the success of a template attack, it is outside the scope of this tutorial to dive into how an adversary will perform the selection of the points-of-interest. The figure below represents a selection of a power trace, of an AES algorithm. The red points in the figure below, represents the selected point sample we use for computing the template.



A *template* h_i is pair which consists of a mean value and a description on variance.



Template matching

The typical explanation of these attacks tends to swamp the brave hart who attempts to decipher them with formulae. We aim to abolish the heavy math description and settle for a more intuitive understanding of this topic.

[xxx] Blue book