Chapter 5

SIMPLE POWER ANALYSIS

Simple power analysis (SPA) attacks are characterized by Kocher *et al.* in [KJJ99] in the following way: "SPA is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations." In other words, the attacker tries to derive the key more or less directly from a given trace. This can make SPA attacks quite challenging in practice. Often, they require detailed knowledge about the implementation of the cryptographic algorithm that is executed by the device under attack. Furthermore, if only one power trace is available, usually complex statistical methods have to be used in order to extract the signal.

SPA attacks are useful in practice if only one or very few traces are available for a given set of inputs. Consider for example a scenario where a consumer uses a smart card to pay for gas at a gas station. The customer has to refill the gas tank of the car on a regular basis and always buys a similar amount of gas. A malicious smart card reader could record the power consumption of the card. In this way, the attacker could gather a couple of traces for similar plaintexts.

In this chapter, we discuss different types of SPA attacks, like the visual inspection of power traces, template-based SPA attacks and collision attacks. The examples for attacks that we provide in this chapter have been produced with the AES software implementation that we describe in Appendix B and the measurement setup that we describe in Section 3.4.4.

5.1 General Description

The goal of SPA attacks is to reveal the key when given only a small number of power traces (for a small number of plaintexts). In the most extreme case, this means that the attacker attempts to reveal the key based on one single power trace. In order to distinguish between the extreme and the normal SPA assumption, we distinguish between *single-shot SPA attacks* and *multiple-shot*

SPA attacks. In single-shot SPA attacks, only one power trace can be recorded. In multiple-shot SPA attacks, multiple power traces can be recorded.

In multiple-shot SPA attacks, either we can measure the power consumption for the same plaintext multiple times, or we can even supply different plaintexts. The advantage of having several traces for one plaintext is that we can use them to reduce the noise by computing the mean of the traces.

Despite the differences in taking a single measurement or taking multiple measurements, the principle of SPA attacks is always the same. The attacker needs to be able to monitor the power consumption of the device under attack. In the attacked device, the key must have (directly or indirectly) a significant impact on the power consumption.

SPA attacks exploit key-dependent differences (patterns) within a trace. They use only one trace or very few traces.

5.2 Visual Inspections of Power Traces

Every algorithm that runs on a cryptographic device is executed in a sequential manner. The operations that are defined by the algorithm are translated into instructions that are supported by the device. For example, AES consists of ten different steps, which are called rounds. Each round consists of four round transformations, which are called AddRoundKey, SubBytes, ShiftRows, and MixColumns. Each round transformation works on one or several bytes of the AES state, see Appendix B.

Suppose that AES is implemented in software on a microcontroller. In this case, the round functions are implemented using the instructions of the microcontroller. Microcontrollers have an instruction set that typically consists of arithmetic instructions (such as addition), logical instructions (such as exclusive-or), data transfer instructions (such as move), and branching instructions (such as jump). Each instruction works on a number of bytes and involves different components of the microcontroller, such as the arithmetic-logic unit, memory (external or internal RAM or ROM), or some peripheral (such as a communication port). These are physically separate components of the microcontroller and they differ in functionality and implementation. Therefore, they have a characteristic power consumption, which leads to a characteristic pattern in the power trace. For instance, a move instruction that operates on data that is stored in internal memory needs fewer clock cycles than a move instruction that operates on values that are located in external memory. Furthermore, the external buses often cause a higher power consumption than the internal buses. These facts make it possible to distinguish instructions in a power trace.

The possibility to distinguish instructions within a trace can lead to a serious security problem if the sequence of instructions directly depends on the key.