Side-Channel Attacks

A third and entirely different family of attacks are side-channel attacks. They exploit information about the private key which is leaked through physical channels such as the power consumption or the timing behavior. In order to observe such channels, an attacker must typically have direct access to the RSA implementation, e.g., in a cell phone or a smart card. Even though side-channel attacks are a large and active field of research in modern cryptography and beyond the scope of this book, we show one particularly impressive such attack against RSA in the following.

Figure 7.4 shows the power trace of an RSA implementation on a microprocessor. More precisely, it shows the electric current drawn by the processor over time. Our goal is to extract the private key d which is used during the RSA decryption. We clearly see intervals of high activity between short periods of less activity. Since the main computational load of RSA is the squarings and multiplication during the exponentiation, we conclude that the high-activity intervals correspond to those two operations. If we look more closely at the power trace, we see that there are high activity intervals which are short and others which are longer. In fact, the longer ones appear to be about twice as long. This behavior is explained by the square-and-multiply algorithm. If an exponent bit has the value 0, only a squaring is performed. If an exponent bit has the value 1, a squaring together with a multiplication is computed. But this timing behavior reveals immediately the key: A long period of activity corresponds to the bit value 1 of the secret key, and a short period to a key bit with value 0. As shown in the figure, by simply looking at the power trace we can identify the secret exponent. Thus we can learn the following 12 bits of the private key by looking at the trace:

operations:	*S*	*SM*	*SM*	*S*	*SM*	*S*	*S*	*SM*	*SM*	*SM*	*S*	*SM*	
private key:	0	1	1	0	1	0	0	1	1	1	0	1	

Obviously, in real-life we can also find all 1024 or 2048 bits of a full private key. During the short periods with low activity, the square-and-multiply algorithm scans and processes the exponent bits before it triggers the next squaring or squaring-and-multiplication sequence. Notice that because the square-and-multiply algorithm doesn't do any operation for the first bit it does not appear in the power trace. However, the first bit is the always 1, therefore the actual private key in binary would be: $(1011010011101)_2$.f

This specific attack is classified as simple power analysis or SPA. There are several countermeasures available to prevent the attack. A simple one is to execute a multiplication with dummy variables after a squaring that corresponds to an exponent bit 0. This results in a power profile (and a run time) which is independent of the exponent. However, countermeasures against more advanced side-channel attacks are not as straightforward.

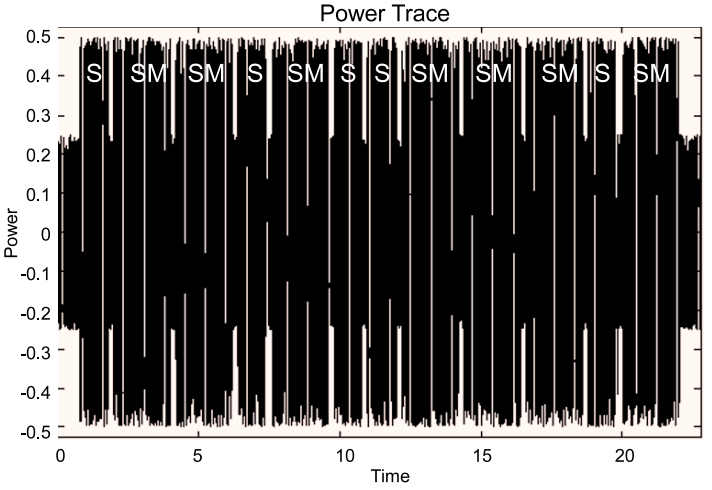

Fig. 7.4 The power trace of an RSA implementation

This is a text based on a chapter from the book Understanding Cryptography by Christof Paar and Jan Pelzl. The book was originally published by Springer.