Report Lab 2: Hardware Security
Diffenretial Power Analysis

Henning Schei

May 2016

1 Introduction

The implementation of the permutation code, `p_permutation`, is a good target for a timing attack, because there is a strong dependence between the hamming weight of the value to be permuted and the running time used. This is because of the aditional 32 iteration loop that the permutation code enter if a bit is set.

From an attackers point of view, this can be used to build up a timing model where the correlation between hamming weight and time consumed are beeing exploited.

2 The algorithm

This is a breif overview of the operations of the algorithm:

- Calculate the output of all 8 S-boxes simutainiously with all possible keys, using an XOR operation with hexadecimal value of += `0x041041041041`.
- Used a masking pattern of `0xf0000000` to isolate each S-boxes bits and shifted the masking pattern appropriate to each S-boxes output.
- A multithreaded function calculated the hamming weight of all 8 S-boxes. The results with hamming weight HW = 0, 1 got stored in a *fast* list and those with HW=3,4 got stored in a *slow* list, respectivly. Based on this, the maximum timing difference between the average time for the slow and fast list on each sbox were used to estimate the most likely subkey.

3 Discussion