Optimization one

To further develop the approach of this algorithm, optimizations were added to make the rate of key discovery more efficient. The first attempt to optimize this program was to examine the effects of different plaintexts on the key discovery. The inspiration for this approach came from the DPA Contest Hall of Fame, where a category for this approach exists, but is void of applicable approaches. To examine the effects of the plaintext on the traces, the plaintexts of several hundred traces are compared to an arbitrary plaintext for similarities on the bit level. The idea is that these similarities at the bit level translate into similar round text after the initial permutations. These similarities are then ordered from greatest to least. Much time was spent analyzing the results of this sort and it was discovered that sorting the results with by varying between the smaller and greater similarities yielded marginal results. This method was compared to the baseline implementation using the same trace chunks and showed a slightly smaller average number of traces.

More efficient methods of anylising this plaintexts could be used to better correlate them one to another. The challenge lies in finding plaintexts that are generally more or less different from every other. Given enough time and a reliable sorting algorithm, this approach could possibly be more effective.

Optimization 2

Optimization Two proved to be much more effective than the first. It includes again sorting the traces but by the amplitude of the first round signal in the trace. The search for this optimization began with observing the effects that different traces had on the rate at which subkeys were discovered. The hypothesis that the rate of key discovery was connected to the type of traces being received was again tested with greater success. Since this implementation analyzed the first round, this round served as the basis of the sort. To perform the sort, the first round of all the traces being sorted were analyzed near the peak of the power impulse. The data points near the peak were then averaged to find a unique number relating to the amplitude. Using this data, the traces were sorted in several patterns until the final sort was found and determined to be most effective. The final method included alternating from the largest amplitude to the smallest, then the next largest, and so on… This order lead to the maximum rate of discovery of on average 180 traces.

The reason this type of sorting is so effective is that the amplitude of the first round is connected to the hamming distance of the round. By using traces that have great variance in the hamming distance, the program is better able to correlate the results to find the subkey and generate key.