Zak Rowland
Assignment 7
Embedded System Security
Topic: AES-128 cracking and attack mitigation

Part 5: Questions

1. From the first page, please answer:

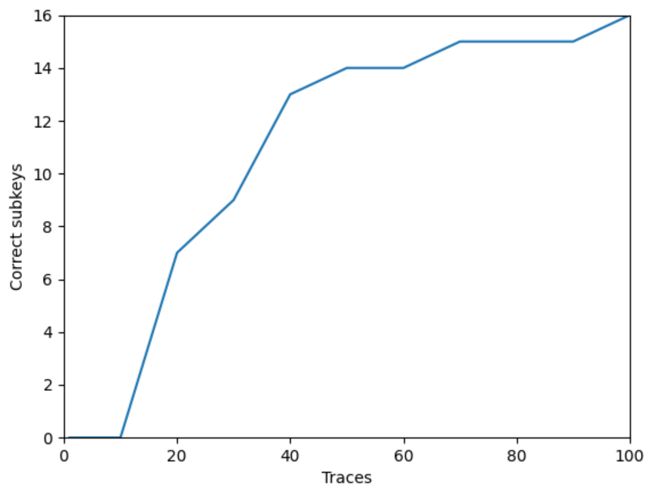
   Run the cw_aes_guess.py file against the sample files on the course website in traces.zip. Unzip the traces.zip file into the same directory as cw_aes_guess.py.

   Record your encryption key that you receive here.

   2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

2. Plot a graph of the # traces vs. correct # of subkeys using **Python.** Use the following number of traces. 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100.

   Insert the plot here.
   (https://matplotlib.org/tutorials/introductory/pyplot.html)



   Based on the graph plots, what is the minimum number of traces required to get the correct key guess for all subkeys?

   Minimum number of traces for all subkeys correct:

   _100_____________________________________

Zak Rowland
Assignment 7
Embedded System Security
Topic: AES-128 cracking and attack mitigation

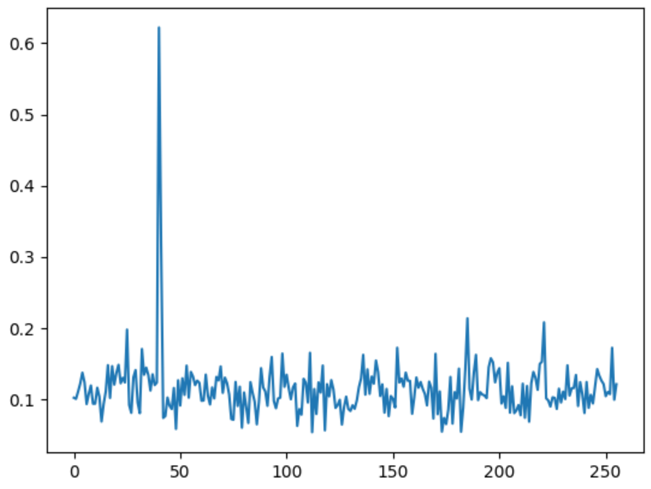
3. What is the PGE, and what is it useful for?

   PGE stands for Partial Guessing Entropy and it is a measure of how many values had a higher correlation coefficient than the actual correct value. In other words, it is the number of incorrect guesses ranked over the correct guess. It is useful in determining if the guess what correct, as indicated by a PGE of zero.

4. If the PGE is non-zero, does it mean that the subkey is protected well?

   Not necessarily. As seen from the test above, at a certain number of traces (about 20,) the PGE begins to rapidly approach zero for all subkeys.

5. Provide a graph in Python of the PGE plot for the 5th subkey.



6. Given the sample capture, what is the minimum number of traces you need to have all subkeys correct? (The cw_aes_guess.py script has a numtraces, where you can adjust the number of traces without re-running the whole capture.)

   Around 100.

Zak Rowland
Assignment 7
Embedded System Security
Topic: AES-128 cracking and attack mitigation

7. We sampled (or would have sampled) the signal at 2mV/div. If we increase to 4mV/div or 8mV/div, do you expect an increase or decrease in number of traces required to correctly guess all subkeys?

    If traces are kept the same, as well as samples per trace, entropy will increase as the volts per division increases. This means traces would have to increase along with the vertical scale in order to continue correctly guessing subkeys.

8. Signal to noise ratio is important in this attack. What methods might you use (hardware or software) to improve the signal to noise ratio available for the attack?

    To improve SNR, use the smallest volts per division and the maximum number of traces and samples per trace possible. Another way is to use a high quality external power supply and oscilloscope.

9. Reviewing the code in cw_aes_guess.py, why are we using the SBOX? What special relationship makes it easier to attack?

    We are using the SBOX because it helps require fewer samples, and this particular SBOX is used in most AES implementations. The SBOX also leaks a lot of information out of the power and EM side channels. The hamming weight and distance in relation to the SBOX is used to make the guess.

10. We are attacking subkeys which are 8 bits wide. What would you anticipate would have to change if we were trying to attack a subkey that was 16 bits wide?

    a. Would the number of captures increase or decrease? Why?

        I'm not sure, I think the captures would have to increase because there are now 65,536 guesses for each subkey instead of 256.

Zak Rowland
Assignment 7
Embedded System Security
Topic: AES-128 cracking and attack mitigation

b. Would we need to collect at a higher or lower sampling rate? Why?

Also not sure, I think it would need to increase as well.

11. If you were asked to perform the same attack on an FPGA or GPU, how might your attack strategy differ?

    To attack something else like an FPGA or GPU, the hamming weight or distance would not be used. They would need to be attacked asynchronously. An FPGA can also change the circuit in real-time making attacks more difficult. A very high signal to noise ratio may also be required.

12. If you were an attacker who did not have a trigger signal, how might you go about synchronizing to the required point in time where the SBOX stage starts? (See Figure 3)

    The SBOX stage starts when the plaintext byte is XOR'd with the key byte, so monitor for this operation.

13. The feasibility of this attack is limited in the field (such as a supermarket or gas station.). What is the usefulness in this attack, and what type of person or organization may want to implement this attack?

    This type of attack is useful for security research, in a lab environment with the victim device, or as insider attack by someone with previous knowledge of the code/implementation.