EECE_5699 Computer Hardware and System Security

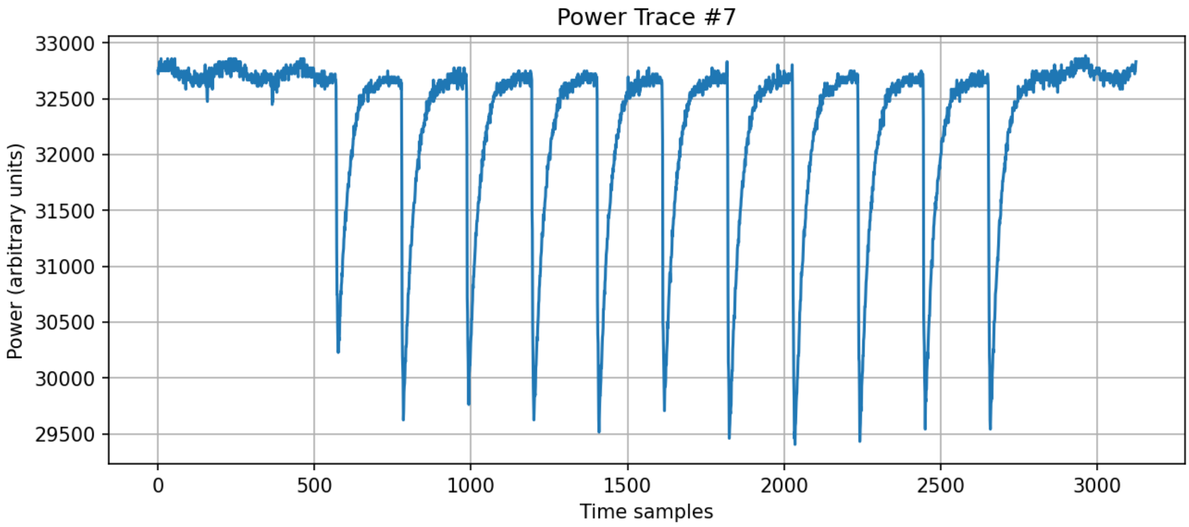
Homework 2

Sakthivel P. Sivakumar

NUID: 002312294

Q1: Generate a plot of one power trace in the data set.

In the given handout we have a total of 7000 Power Traces of which I can choose any one for demonstrating the Plot between the Time Samples (in X axis) and Power Consumption (in Y axis) For my demo I have chosen the Power Trace **No: 7**



Q1.1: why there are 11 dips instead of 10 dips?

AES – 128 Structure: 128-bit AES has 10 rounds of encryption, however the no. of round keys in total is 11 because the formula for Total Number of Round Keys is

Total Number of Round Keys = No. of Rounds + 1

The additional round key is used for the initial AddRoundKey operation before the main rounds (1-10) begin. The Key Expansion algorithm generates all round keys from the original encryption key. Each dip in the power trace corresponds to an AddRoundKey operation, which is the most power-consuming operation in AES due to the XOR operations across all 16 bytes simultaneously.

- **Dip 1:** Initial AddRoundKey before Round 1
- **Dip 2 to 10:** AddRoundKey operations at the end of Rounds 1-9
- **Dip 11:** AddRoundKey operation in the final Round 10

In Dip 1 we have just the AddRoundKey operation, In Dip 2 to 10 we have 4 transformations in each round **(substitute bytes → Shift Rows → Mix Columns → AddRoundKey**). In Dip 11 (round 10), we have just 3 transformations (Mix Column is excluded and the remaining transformations occur).

Q2: Correlation Power Analysis with Hamming Distance Power Model

Objective:

In this part of the assignment, I have implemented the Correlation Power Analysis Attack on a 128-bit AES using the Hamming Distance power model to recover the last round key.

Methodology:

1) Hamming Distance Power Model – The attack targets the last round of AES encryption and uses the relationship

$$\mathbf{Cj = sbox[Si] \oplus Kj}$$

Due to the ShiftRows operation, there's a mapping between input state index 'i' and output state index 'j'.

2) For each key byte guess, we calculate:

$$\mathbf{Si = inv\_sbox[Cj \oplus Kj]}$$

3) Hamming Distance

hd = Hamming_Distance(Si, Ci)

After this, we correlate the predicted power values (HD) with actual power measurements at the leakage point

correlation = pearsonr(hd_values, power_traces)

Implementation:

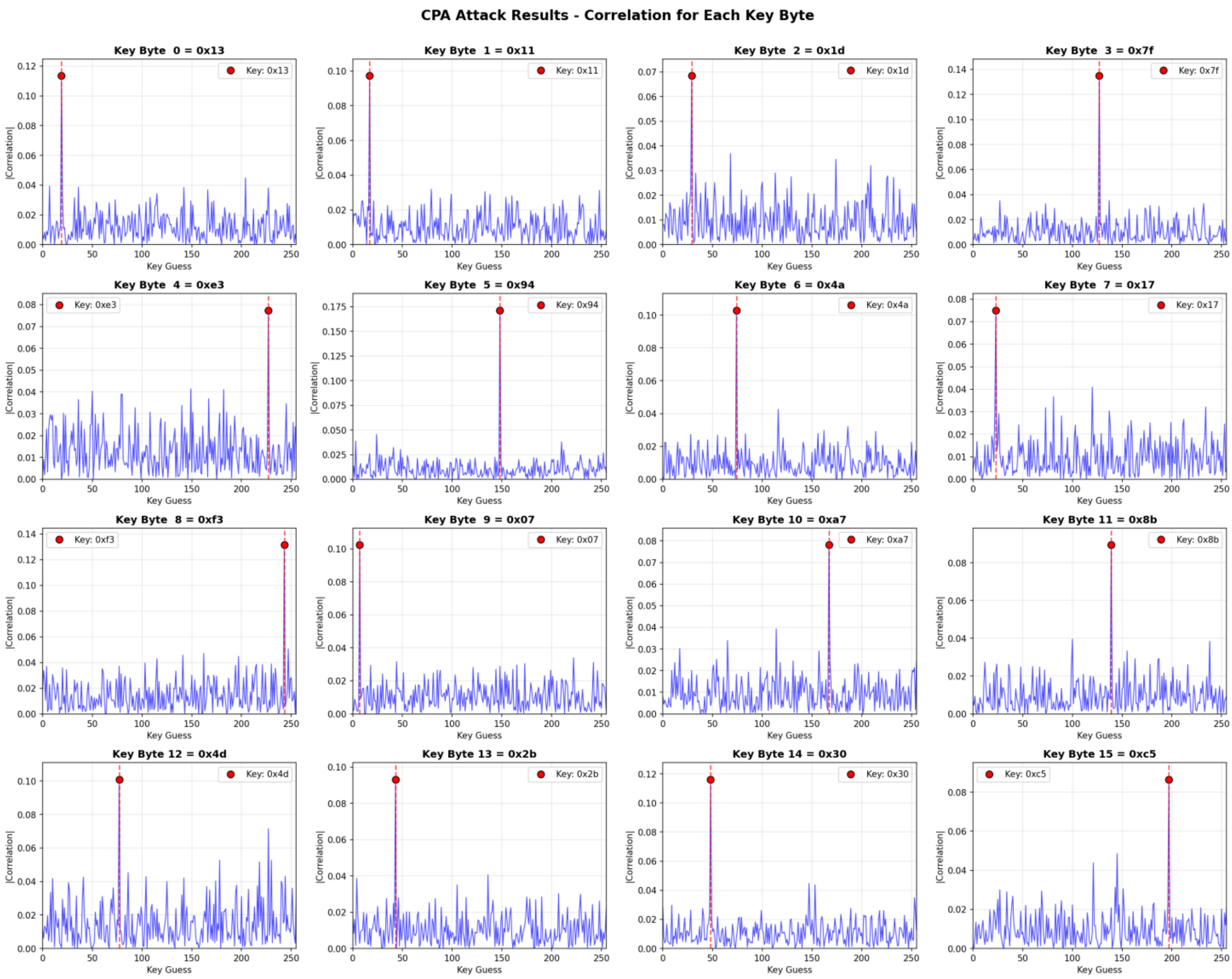
- First set up the necessary Python Environment and install the necessary packages (**matplotlib, numpy and scipy** etc.)
- Then run the **CPA_attack.py** code

CPA Attack Results:

```
Recovered AES Key:
0x13 0x11 0x1d 0x7f 0xe3 0x94 0x4a 0x17 0xf3 0x07 0xa7 0x8b 0x4d 0x2b 0x30 0xc5
Key: 13111d7fe3944a17f307a78b4d2b30c5
Attack successful!
```

The Recovered AES key in Hex is:

13	**11**	**1d**	**7f**	**e3**	**94**	**4a**	**17**	**f3**	**07**	**a7**	**8b**	**4d**	**2b**	**30**	**c5**



Correlation Analysis:

```
Max correlation values for each key byte:
Key byte  0: 0.1135
Key byte  1: 0.0973
Key byte  2: 0.0684
Key byte  3: 0.1350
Key byte  4: 0.0774
Key byte  5: 0.1713
Key byte  6: 0.1029
Key byte  7: 0.0750
Key byte  8: 0.1314
Key byte  9: 0.1023
Key byte 10: 0.0781
Key byte 11: 0.0895
Key byte 12: 0.1008
Key byte 13: 0.0931
Key byte 14: 0.1161
Key byte 15: 0.0864

Correlation Statistics:
Average: 0.1024
Minimum: 0.0684
Maximum: 0.1713
```

Experience Summary:

By analyzing 7000 power traces and their corresponding ciphertexts, the attack successfully recovers all 16 bytes of the AES key through statistical correlation analysis. The implementation demonstrates how side-channel attacks can exploit physical characteristics (power consumption) to break cryptographic systems. The attack achieves strong correlations (average ~0.10) across all key bytes, confirming successful key recovery. This work highlights the importance of implementing countermeasures against power analysis attacks in real-world cryptographic devices.