SE518 — Lab 1 Report

Security Lab – Introduction to Correlation Power Analysis

SE518 — Embedded Systems Security
Supervisor: Prof. MIRBAHA Amir-Pasha

Group Members

- HAMZA Muhammed
- IBRAHIM Thomas
- DA ROZA Lukas

Lab Questions and Answers

1.1. Explain what each of the arrays: D, K, V, H and R represents.

- **D:** plaintext bytes for attacked byte (`samples × 1`).
- **K:** key hypotheses `0..255` (`256 × 1`).
- **V:** `Plaintext XOR Key` (intermediate) (`256 × samples`).

- **H:** hypothetical leakage (Hamming Weight of S-Box output) (`256 × samples`).
- **R:** correlation coefficients between H and measured traces (`256 × trace_length`).

1.2. Complete the missing code in order to perform the Correlation Power Analysis (CPA) attack.

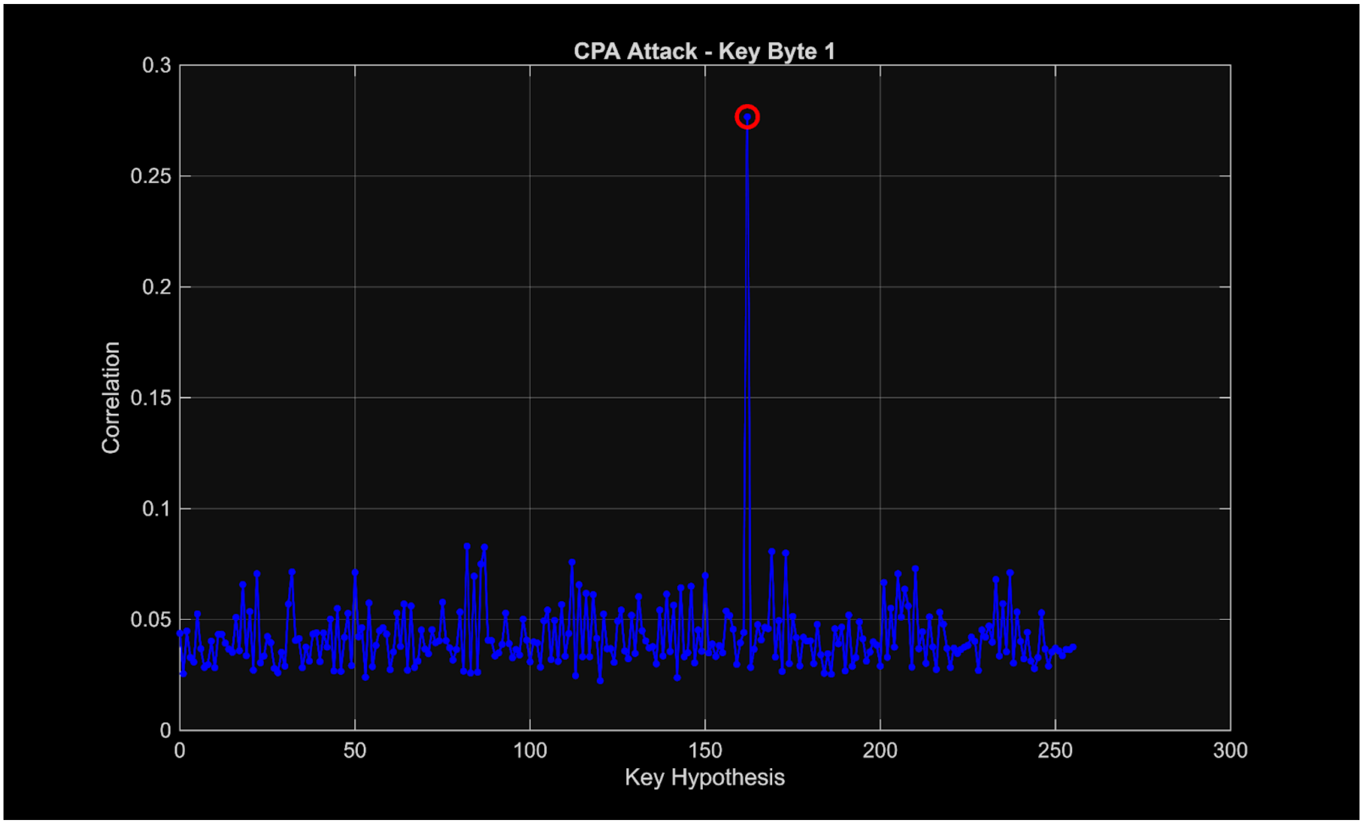
```
 **% Prepare data - get ONLY the current byte we're attacking

   D = plaintexts_SCA(1:samples, byte_to_attack);

   % Prepare keys - all possible byte values (0-255)

   K = 0:255;

   % Calculate hypothetical intermediate values after AddRoundKey

   V = zeros(length(K), samples, 'uint8');

   for key_idx = 1:length(K)

       V(key_idx, :) = bitxor(D, K(key_idx));

   end

   % Calculate hypothetical power consumption using Hamming Weight

   H = zeros(length(K), samples);

   for key_idx = 1:length(K)

       for sample_idx = 1:samples

           % Apply S-Box then get Hamming Weight

           sbox_output = SubBytes(V(key_idx, sample_idx) + 1); % +1 for MATLAB indexing

           H(key_idx, sample_idx) = HW(sbox_output + 1); % Hamming Weight of S-Box output

       end

   end

   % Calculate the correlation
```

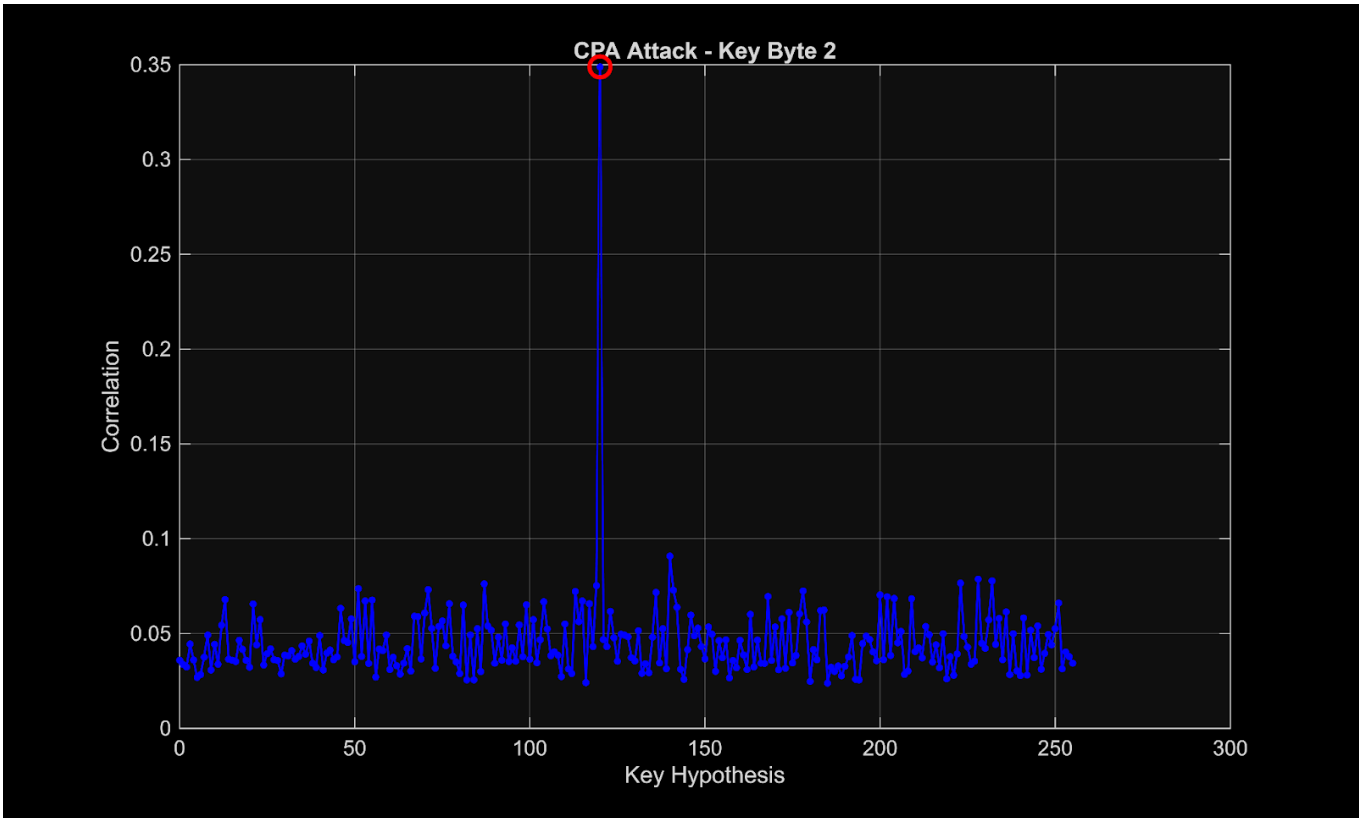
```
    trace_length = size(traces, 2);

    R = zeros(length(K), trace_length);

    for key_index = 1:length(K)

        if (mode == 1 && mod(key_index, 50) == 0)

            fprintf('Working on key guess = %d\n', K(key_index));

        end

        for k = 1:trace_length

            % Calculate correlation between hypothetical power and actual traces

            correlation_matrix = corrcoef(H(key_index,:), traces(:,k)');

            R(key_index, k) = correlation_matrix(1,2);

        end

    end

    [M,I] = max(abs(R(:)));

    [key_row, key_col] = ind2sub(size(R),I);

    key_found = key_row - 1;**
```

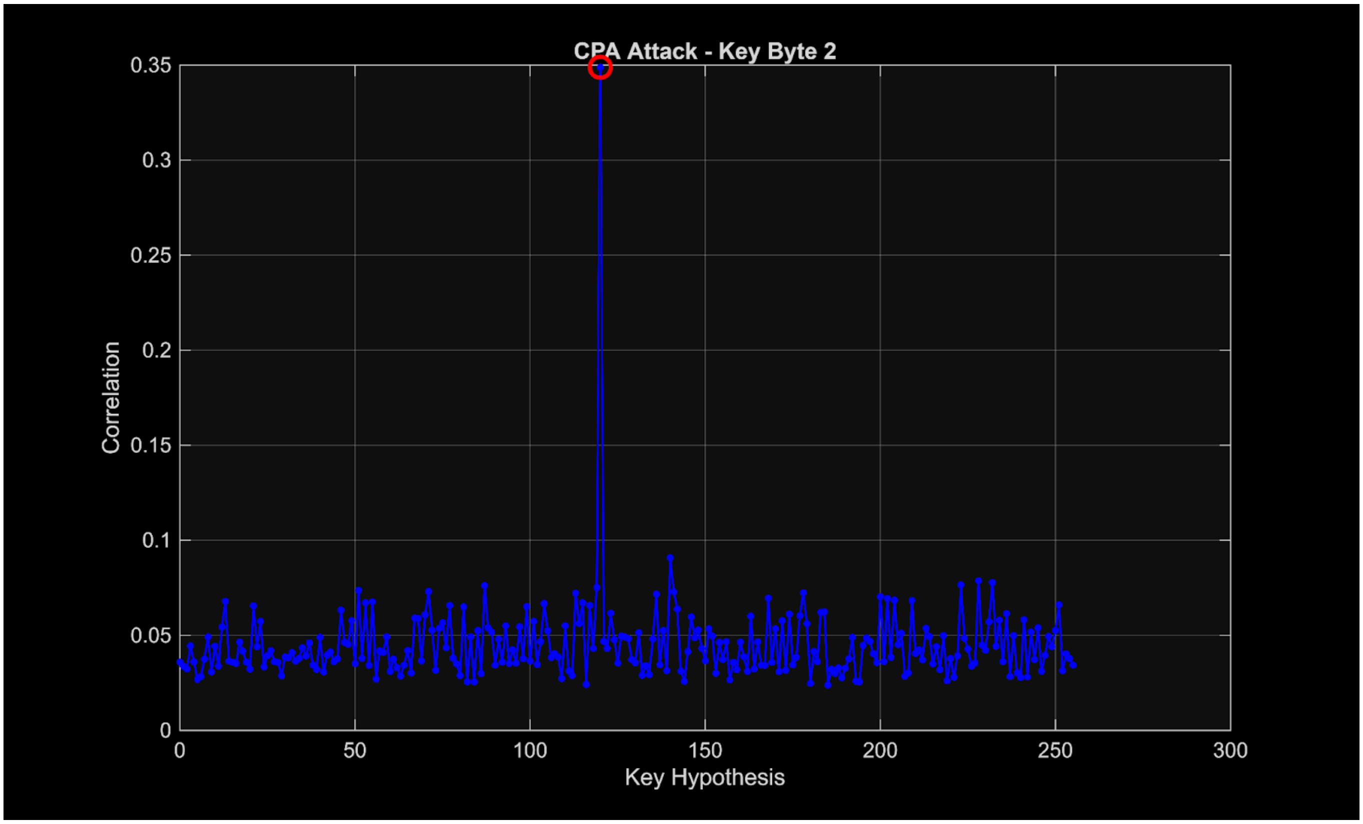
1.4. Perform the CPA attack and try to find only some bytes of the key according to the instructor directions. Include in your report the key you have found.

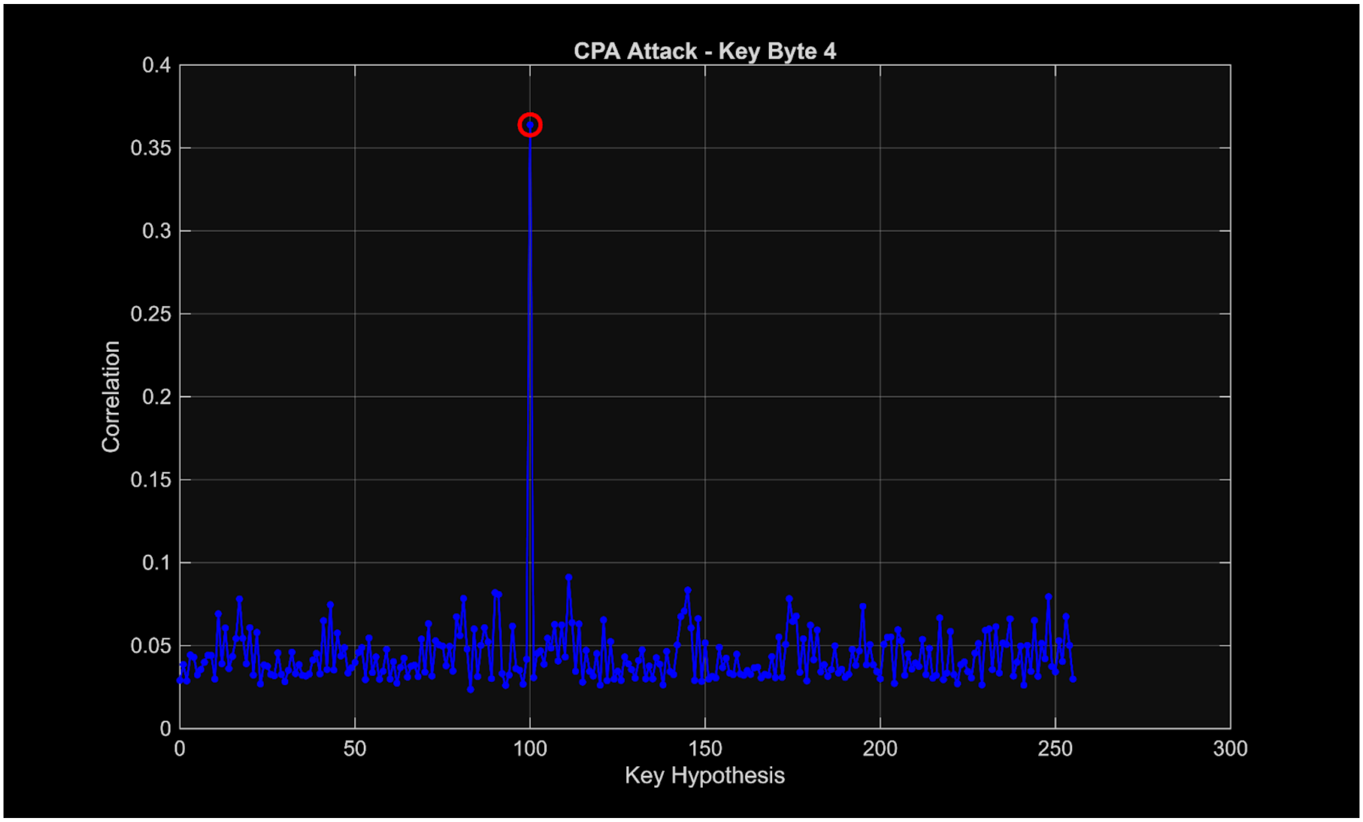
Following are the keys that we found as a result of CPA Attack

```
Byte 1: Found key = 162
Byte 2: Found key = 120
Byte 3: Found key = 91
Byte 4: Found key = 100
```


CPA Attack - Key Byte 1
Correlation
Key Hypothesis
0
0.05
0.1
0.15
0.2
0.25
0.3
0
50
100
150
200
250
300


CPA Attack - Key Byte 2
Correlation
Key Hypothesis
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0
50
100
150
200
250
300





1.5. What is the maximum correlation?

The maximum correlation we found was of 0.397794

Correlation of each byte found is given below:

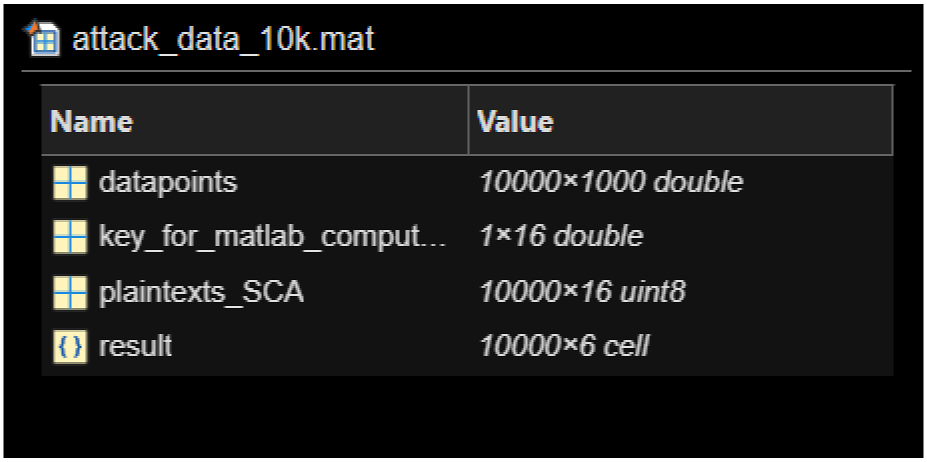
```
Byte 1: Correlation = 0.276665
Byte 2: Correlation = 0.348889
Byte 3: Correlation = 0.397794
Byte 4: Correlation = 0.363957
```

1.6. Explain in your report why, in your opinion, Hamming Weight can be used as a hypothetical power model.

Hamming Weight is an effective power model because:

- The number of '1' bits in a register correlates with power consumption in CMOS circuits
- More bit flips hence more dynamic power consumption
- Easy to compute and correlates well with actual power measurement

1.8 Explain the content of the file “attack_data.mat” in detail.**



attack_data_10k.mat

Name	Value
datapoints	*10000×1000 double*
key_for_matlab_comput...	*1×16 double*
plaintexts_SCA	*10000×16 uint8*
result	*10000×6 cell*

- plaintexts_SCA: It contains the known plaintexts that were encrypted i.e plaintext bytes. Basically, plaintext data for the CPA attack.
- Key_for_matlab_computation_dec: actual AES encryption key byte found
- datapoints: The actual power consumption measurements of 10,000 different encryption traces. This is our T matrix.
- result: probably some miscellaneous results.

1.9. Why is array R of size KxT ?

The correlation matrix R has dimensions K × T (256 × 1000 in this case) because it represents the comprehensive statistical relationship between all possible key hypotheses and all time points in the power measurement.

K = 256: Represents all possible hypotheses for one key byte (0 to 255)
T = 1000: Represents the number of time samples in each power trace

1.10. Minimum Number of Traces

- **Theoretical minimum**: ~100-200 traces for a strong signal
- **Practical minimum**: Usually 500-1000 traces for reliable results
- Depends on **signal-to-noise** ratio and measurement quality

1.11. MATLAB Function for Correlation

The function is **corrcoef**() which calculates the Pearson correlation coefficient matrix.

1.12. Ways to Increase Security Against CPA

1. Masking

Add random masks to intermediate values. It makes power consumption independent of actual data
Example: Boolean masking on S-Box inputs/outputs

2. Hiding Techniques

- **Noise injection**: Add random power consumption
- **Shuffling**: Randomize operation order
- **Balanced logic**: Use circuits with constant power consumption

3. Other Countermeasures

- **Voltage/clock randomization**: Vary operating conditions
- **Secure S-Box implementations**: Use masked S-Boxes
- **Time randomization**: Add random delays between operations