Correlation Power Analysis Attack on AES-128 Encryption

Last-Round Key Recovery Using Hamming Distance Leakage Model

Shubham Rawat
251110060

Sankalp Sharma
251110063

Kajal Kumari
251110040

Divakar Bajpai
251110029

Venkatesh Kumar
251110617

Abstract

This report presents a comprehensive implementation and analysis of a Correlation Power Analysis (CPA) attack targeting the Advanced Encryption Standard (AES-128) cryptographic implementation. The attack exploits side-channel power consumption leakage during the last round of AES decryption to recover a single byte of the 10th round key. We employ the Hamming Distance (HD) leakage model, which captures the power consumption characteristics arising from bit transitions in hardware registers. Utilizing a dataset of 100,000 power traces containing plaintext-ciphertext pairs and corresponding power measurements, we successfully recovered byte 10 of the round key with value `0xC0`.

1 Detailed procedure

1.1 Dataset Description

The experimental dataset consists of power traces collected during AES-128 encryption operations:

- **Total Traces**: 100,000 encryptions
- **Samples per Trace**: 1,249 time points
- **Data Format**: CSV file with columns:
  - Column 0: Plaintext (32 hex characters)
  - Column 1: Ciphertext (32 hex characters)
  - Columns 2-1250: Power measurements (floating-point values)
- **Target**: Byte 10 of the 10th round key

1.2 Hypothesis Matrix Construction

The `build_H` function generates the hypothesis matrix $H \in \mathbb{R}^{M \times 256}$:

Algorithm 1 Hypothesis Matrix Construction (Hamming Distance Model)

Input: Ciphertext bytes $c \in \mathbb{R}^D$, State bytes $s \in \mathbb{R}^D$, Number of traces D
Output: Hypothesis matrix $H \in \mathbb{R}^{D \times 256}$
Initialize H as zero matrix of size $D \times 256$
for each key guess $k = 0$ to 255 **do**
    for each trace $i = 0$ to $D - 1$ **do**
        $V \leftarrow c[i] \oplus k$ ▷ XOR ciphertext with key guess
        $V \leftarrow \text{InvSBox}[V]$ ▷ Apply inverse S-box
        $V \leftarrow s[i] \oplus V$ ▷ Hamming Distance: XOR with previous state
        $H[i, k] \leftarrow \text{HammingWeight}(V)$ ▷ Count number of 1-bits
    end for
end for
return H

1.3 Correlation Computation Algorithm

The `run_cpa` function computes correlations efficiently using matrix operations:

1.4 Byte Index Computation

For AES-128, ciphertext bytes are represented as 32-character hexadecimal strings. To extract byte b:

$$\text{byte\_position} = 2b : 2b + 2$$

For byte 10, accounting for inverse ShiftRows transformation:

```
inv_shift = [0,5,10,15,4,9,14,3,8,13,2,7,12,1,6,11]
target_byte = 10
shifted_position = inv_shift[target_byte]  # Position 10
# Extract from hex string: positions 20:22
```

2 Implementation Details

2.1 Attack Execution

```
# Build hypothesis matrix using HD model
H_hd = build_H(c_bytes, usedD, s_bytes, model="HD")

# Run CPA attack
guess_hd, corr_hd = run_cpa(H_hd, T)

# Compute maximum correlation
max_corr_hd = np.max(np.abs(corr_hd))

# Display results
print(f"Byte {target_byte:2d}: HD guess = {guess_hd:3d}")
print(f"HD guess = 0x{guess_hd:02X}, "
      f"max |corr| = {max_corr_hd:.4f}")
```

2.2 Computational Complexity

The algorithm's time complexity is dominated by matrix multiplication:

- **Hypothesis Construction**: $O(M \times 256)$ where $M = 100,000$
- **Correlation Computation**: $O(256 \times M \times N)$ where $N = 1,249$
- **Total Operations**: Approximately 3.2×10^{10} floating-point operations

NumPy's optimized BLAS routines enable efficient execution on standard hardware.

3 Experimental Results and Conclusion

3.1 Key Recovery Results

The CPA attack successfully recovered byte 10 of the AES-128 round key:

Table 1: Attack Results Summary

Parameter	**Value**	**Unit**
Target Byte Index	10	-
Recovered Key Value (HD Model)	0xC0	hex
Decimal Value	192	dec
Maximum Absolute Correlation	0.0320	-
Number of Traces Used	100,000	traces
Samples per Trace	1,249	samples

3.2 Correlation Trace Analysis

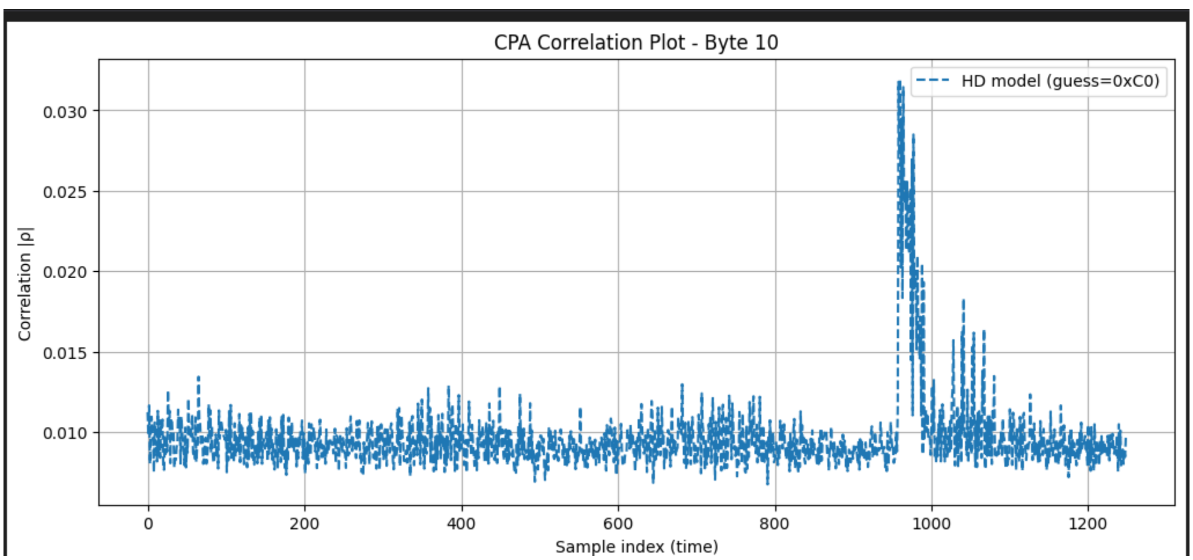

Figure 1: Correlation vs Sample

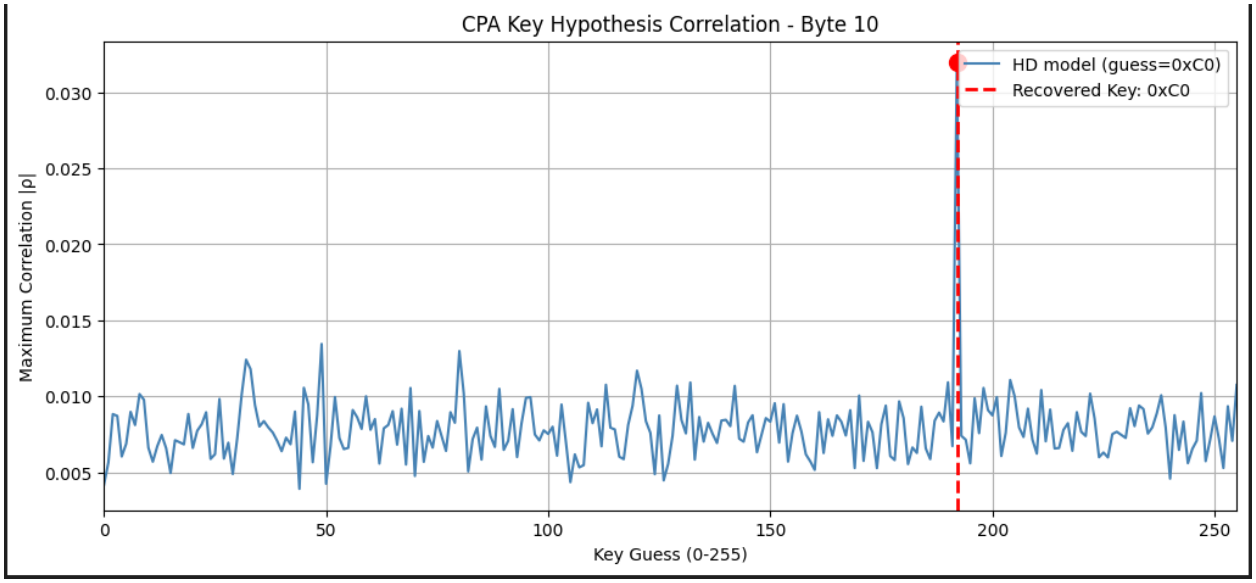

Figure 2: Key Prediction through correlation