Power Analysis Part VI: 2nd Order DPA

Cryptographic Hardware for Embedded Systems
ECE 3170

Fall 2025

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Reading

- This lecture is based on Chapter 10 of *Power Analysis Attacks: Revealing the Secrets of Smart Cards* by Mangard et al., 2007, ISBN-13: 978-0-387-30857-9, ISBN-10: 0-387-30857-1, e-ISBN-10: 0-387-38162-7.

Masking and First-Order DPA

- Note that some types of masking are not 100% resistant to first-order differential power analysis (1^{st}-order DPA)
- Consider multiplicative masking: $v_m = v * m$
- So-called Zero-Value (ZV) power models separate when $v = 0$ from when v is nonzero
- ZV 1^{st}-order DPA attacks may succeed because the power trace for $v = 0$ has no dependence on the mask value

Masking and First-Order DPA (cont'd)

- Consider a combination of Boolean masking with multiplicative masking using the same mask
- For example, consider $(v \oplus m) * m$
  - If $m = 0$, then $v_m = (v \oplus m) * m = 0$
  - Furthermore, $v = 1$, then $v_m = (v \oplus m) * m = 0$ in two cases:
    - $m = 0$
    - $m = 1$
- ZV 1^{st}-order DPA attacks may succeed here as well

Second-order Differential Power Analysis

- Preprocess the data
- In 2nd order DPA, the data is combined in a particular way prior to looking for differences among groups of power traces
  - Assumption is that there are two (or more) intermediate data samples whose corresponding power traces, taken individually, have no known relationship to known data values (e.g., the plaintext input is a known data value)
  - However, a combination of the two or more intermediate data samples does have a known relationship
- Recall that in 1st order DPA, raw power trace values are used directly

Consider the Following Masking Type

- Boolean
  - Intermediate values v and u are concealed (at different points in time and at different parts of the cryptographic algorithm) by XOR with the identical mask m
  - $v_m = v \oplus m$
  - $u_m = u \oplus m$
- Why the same mask?
  - In order to avoid the necessity of the equivalent of a one-time pad, there is some amount of mask reuse in practical implementations

Possible Preprocessing Step

- $w = comb(u_m, v_m) = u_m \oplus v_m$
- Note that $u_m \oplus v_m = u \oplus m \oplus v_m = u \oplus m \oplus v \oplus m = u \oplus v \oplus m \oplus m$ $= u \oplus v \oplus \mathbf{0}$ (where **0** has all zeros and is the same size as u, v and m) $= u \oplus v$
- Therefore $w = comb(u_m, v_m) = u_m \oplus v_m = u \oplus v = comb(u,v)$!!
  - Note that we know the result of $w = u_m \oplus v_m = u \oplus v$ independent of the mask, i.e., without knowing the specific value of m !!!

Preprocessed Traces

- Typically, not certain exactly when u_m and v_m occur
  - E.g., due to shuffling
- However, there is a reasonable interval over which can compare
- E.g., let the interval be $\mathcal{I} = t_{r+1}, \ldots, t_{r+\ell}$ which likely contains u_m and v_m
- Preprocessing compares points t_x, t_y with $x \neq y$
- $(pre(t_{r+1},t_{r+2}), pre(t_{r+1},t_{r+3}),\ldots, pre(t_{r+1},t_{r+\ell}), pre(t_{r+2},t_{r+3}),\ldots, pre(t_{r+\ell-1},t_{r+\ell}))$
- Size of $pre(t_{r+1},t_{r+2}), pre(t_{r+1},t_{r+3}),\ldots, pre(t_{r+1},t_{r+\ell})$ is ℓ-1
- Size of $pre(t_{r+2},t_{r+3}), pre(t_{r+2},t_{r+4}),\ldots, pre(t_{r+2},t_{r+\ell})$ is ℓ-2
- …
- Total length is $(\ell-1) + (\ell-2) + \ldots + 2 + 1 = \ell(\ell-1)/2 = (\ell^2 - \ell)/2$

2^{nd} (and Higher!) Order DPA Attacks

1. Choose an intermediate part of the algorithm to attack
   a. For example, function *f(d,k)* where *d* is a data input and *k* is a small part of the secret key stored in the device under attack
   b. Typically *d* is either plaintext or ciphertext
2. Make a large number of power measurements
   a. Keep track of the known data values d_i as recording the measurements
   b. For each d_i there exists a power trace of size T: $\boldsymbol{t'_i} = (t_{i,1}, \ldots, t_{i,T})$
   c. Preprocess the power traces; for 2^{nd} order, a window of size ℓ adds $\sim\ell^2/2$ preprocessing calculations
3. Calculate hypothetical intermediate values
   a. For each k, the K possible choices are $\mathbf{k} = (k_1, \ldots, k_K)$
   b. The possible choices are used in conjunction with *f(d,k)*
4. Map hypothetical intermediate values to resulting predicted power values
   a. For 2^{nd} order, duplicate the $\sim\ell^2$ preprocessing calculations
5. Compare predicted power values with the actual trace values
   a. Statistical tests involve differences, whether difference of means, covariance, or other

Recall Slide 9 of Lecture 23 Power Analysis Part I:

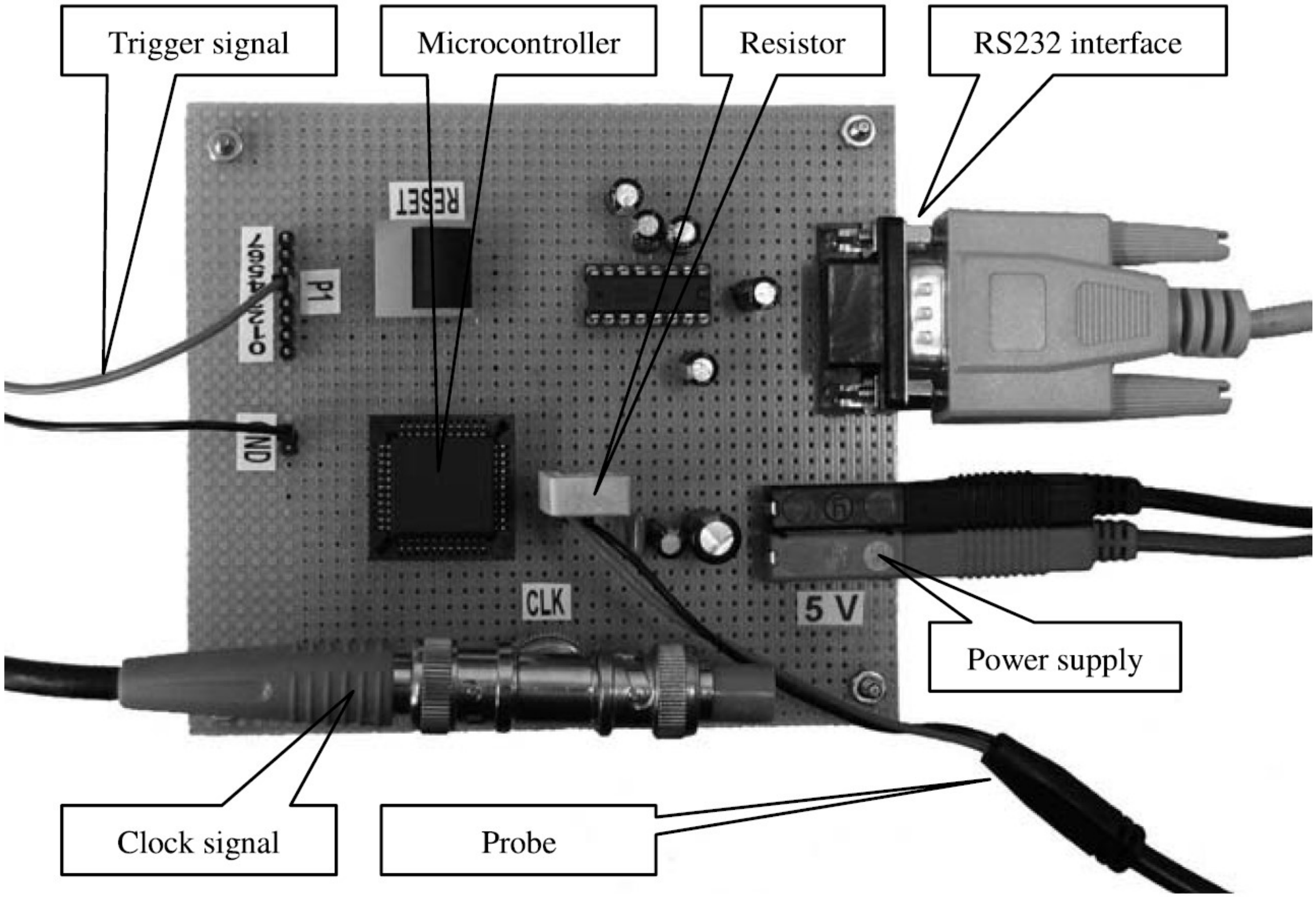
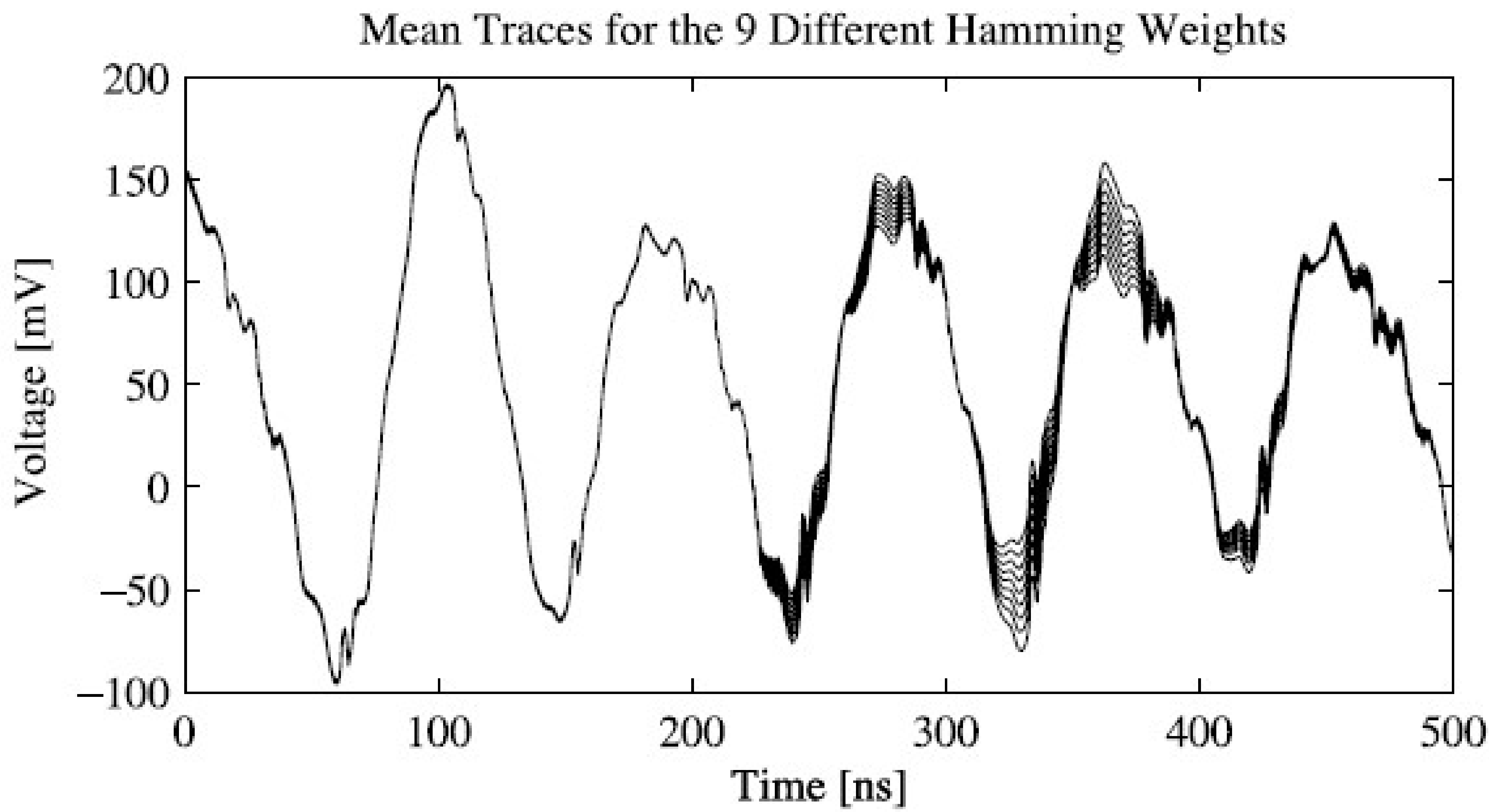


Figure 3.9. Picture of the measurement setup for the attacks on the 8-bit microcontroller.

Recall Slide 38 of Lecture 23 Power Analysis Part I:



Recall Slide 4 of Lecture 24 Power Analysis Part II:

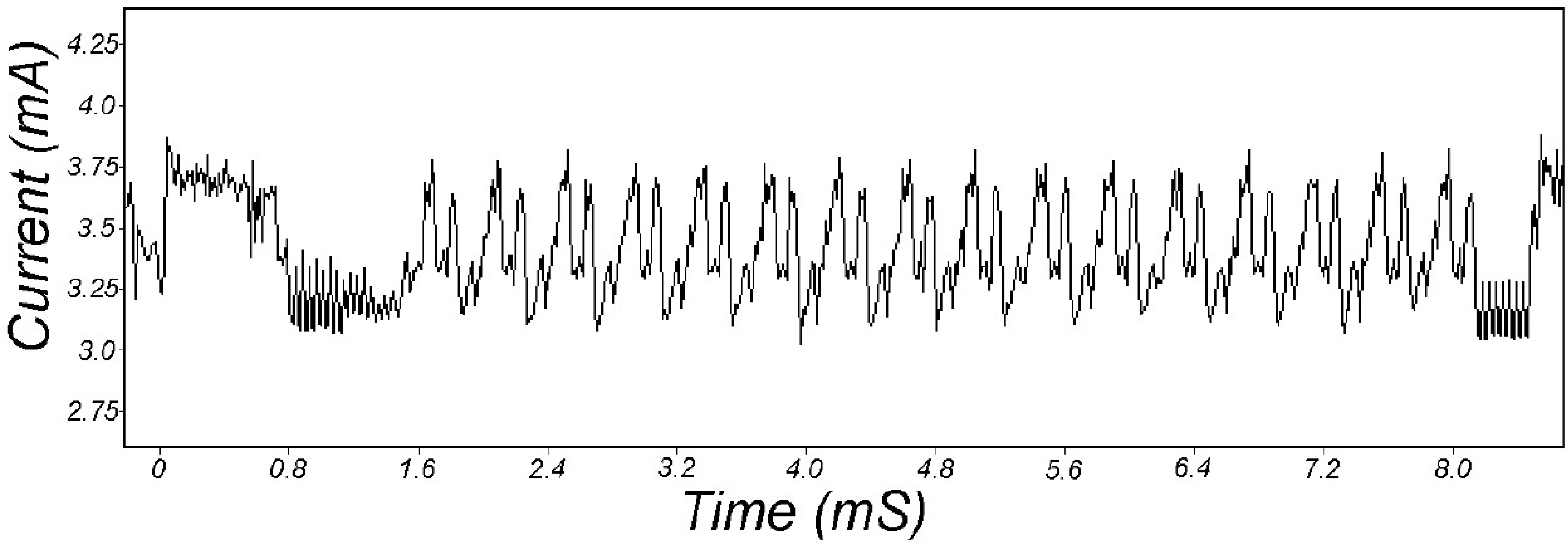


Figure A.1. SPA trace showing an entire DES operation

Recall Slide 6 of Lecture 24 Power Analysis Part II:

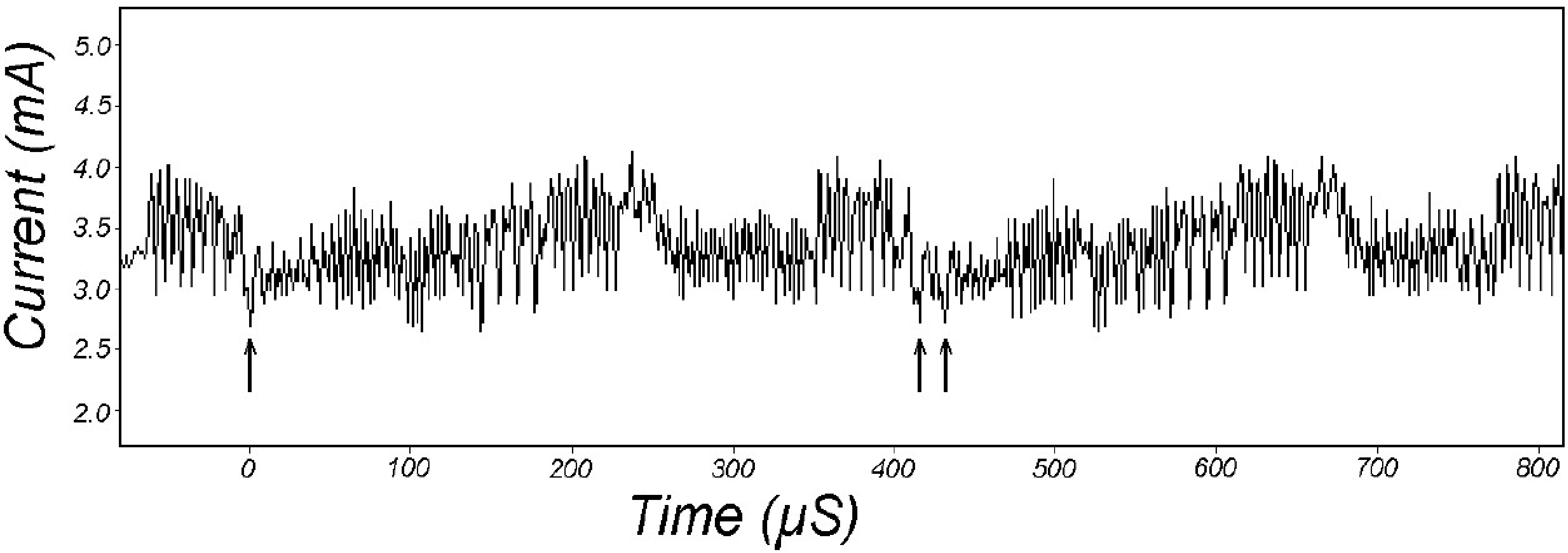


Figure A.2. SPA trace showing DES rounds 2 and 3.

Recall Slide 8 of Lecture 24 Power Analysis Part II:

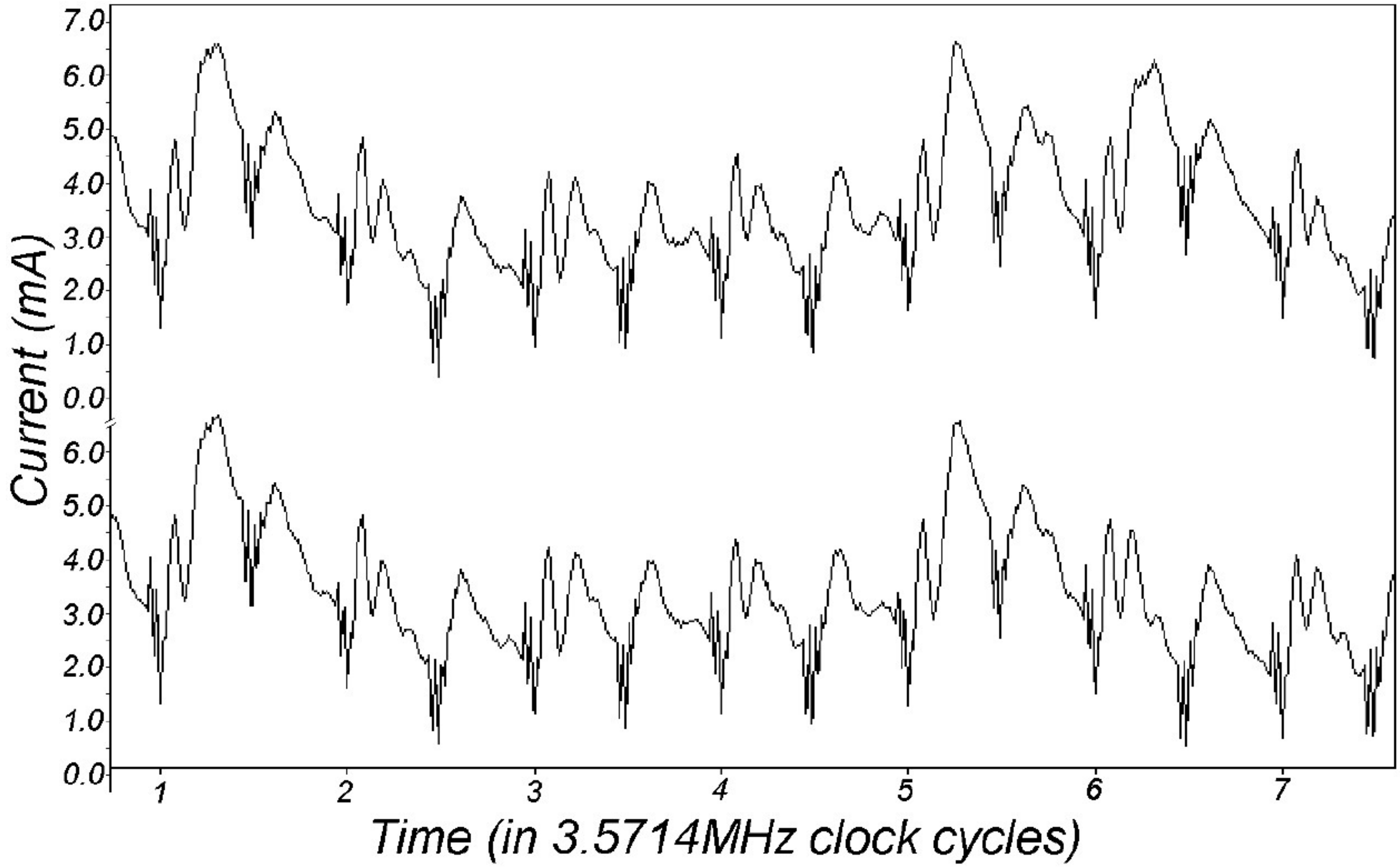


Figure A.3. SPA trace showing individual clock cycles.

Recall Slide 6 of Lecture 28 Power Analysis Part IV:

Steps in Differential Power Analysis

1. Choose an intermediate part of the algorithm to attack
   a. For example, function $f(d,k)$ where d is a data input and k is a small part of the secret key stored in the device under attack
   b. Typically d is either plaintext or ciphertext
2. Make a large number of power measurements
3. Calculate hypothetical intermediate values
4. Map hypothetical intermediate values to resulting predicted power values
5. Compare predicted power values with the actual trace values

Recall Slide 7 of Lecture 28 Power Analysis Part IV:

Difference of Mean Power

- Consider MSB *v* of *S(p ⊕ k)*
  - *v*=1 vs. *v*=0
- Calc. difference of 1000 traces: ½ with *v*=1, ½ with *v*=0
- Note key byte *k* has 256 guesses
- Peaks indicate correct guess!

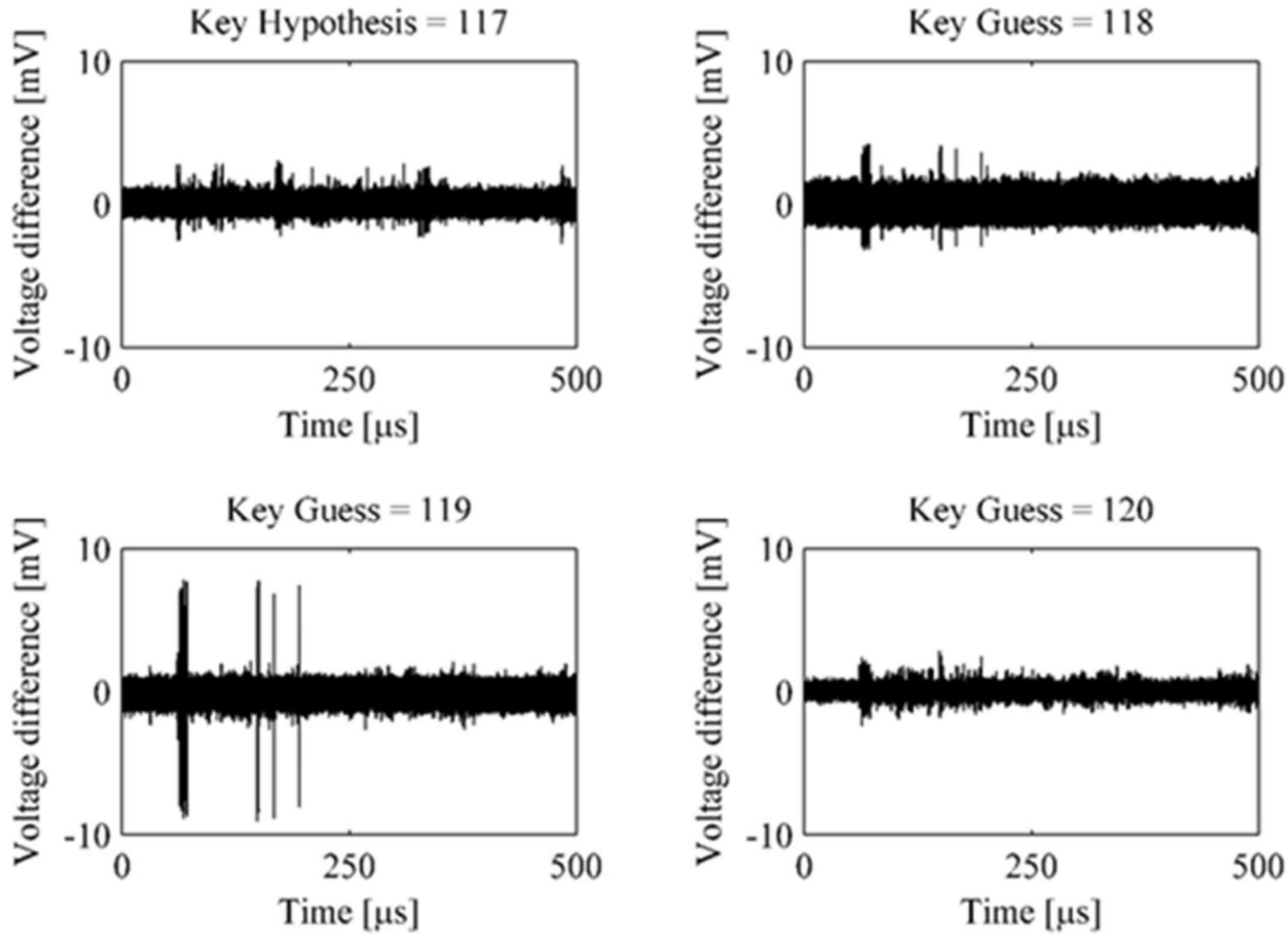


Figure 1.5. Difference plots for the key guesses 117, 118, 119, and 120.

Recall Slide 7 of Lecture 23 Statistics I:

Correlation and Covariance

- Two points are correlated if they vary together in a related way
- Statistical measure: covariance
- *Cov(X,Y) = E[(X-E(X))*(Y-E(Y))] = E(XY) – E(X)E(Y)*
- Theoretical and empirical formulas:
- $\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)*Var(Y)}}$
- $r = \frac{\sum_{i=1}^{n}(x_i - \overline{x_i})*(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x_i})^2 * \sum_{i=1}^{n}(y_i - \overline{y_i})^2}}$
- As defined, the correlation coefficient ρ varies between -1 and 1, i.e., -1 ≤ ρ ≤ 1 and also thus -1 ≤ *r* ≤ 1

Recall Slide 50 of Lecture 21 Power Analysis Part I:

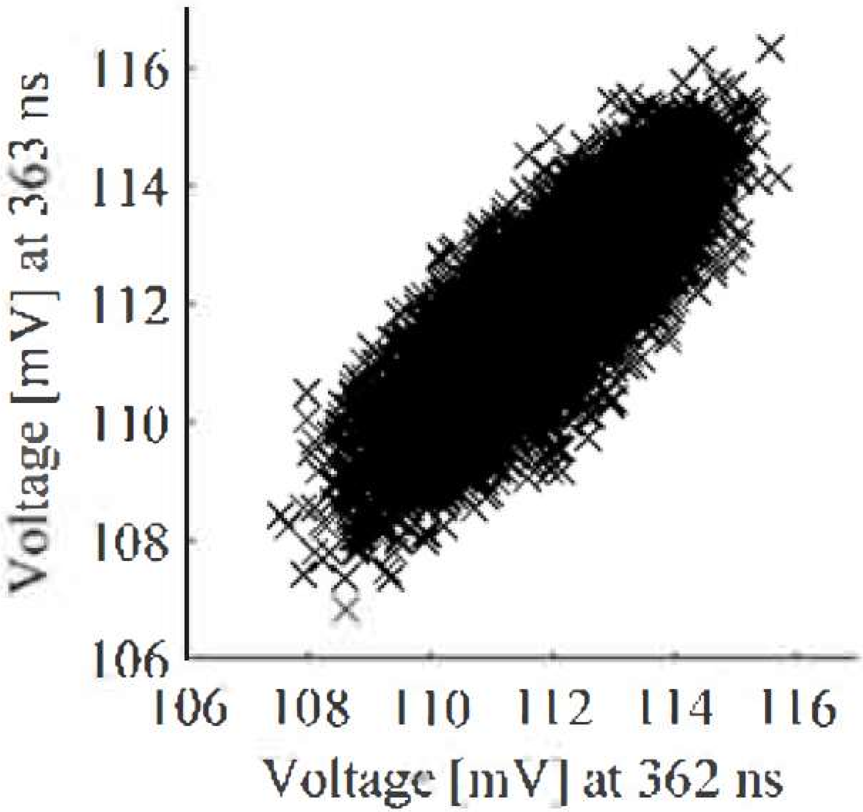


Figure 4.9. Scatter Plot: The power consumption at 362 ns is correlated to the power consumption at 363 ns. ***r*** **= 0.82**

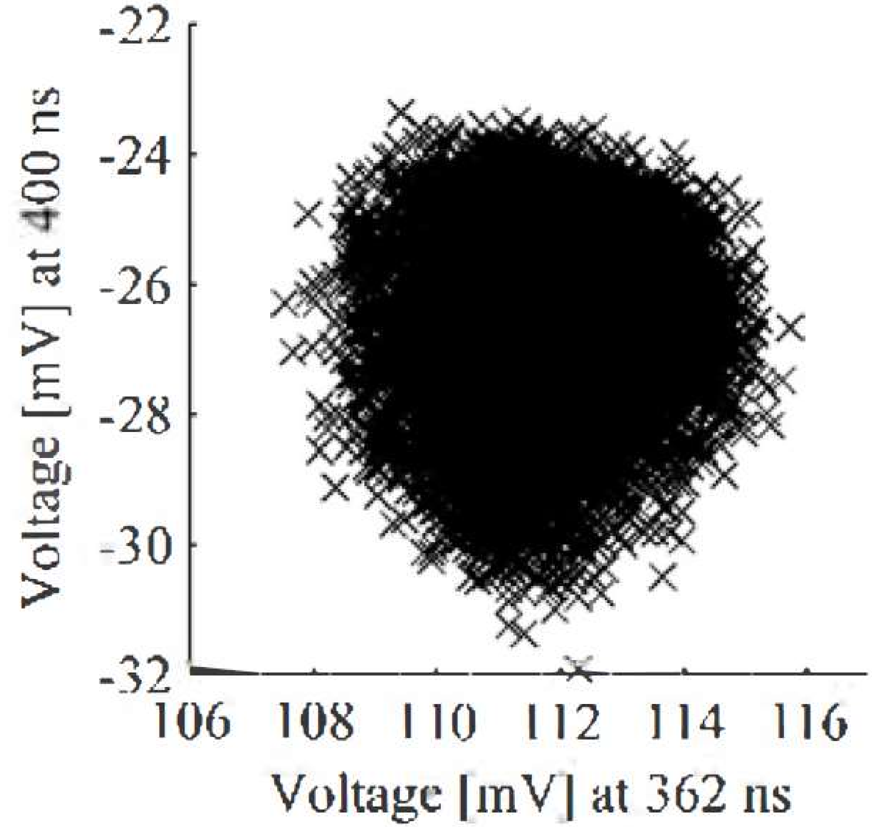


Figure 4.10. Scatter Plot: The power consumption at 362 ns is largely uncorrelated to the power consumption at 400 ns. ***r*** **= 0.12**

Recall Slide 4 of Lecture 29 Power Analysis Part V: Steps in Differential Power Analysis

1. Choose an intermediate part of the algorithm to attack
   a. For example, function $f(d,k)$ where d is a data input and k is a small part of the secret key stored in the device under attack
   b. Typically d is either plaintext or ciphertext
2. Make a large number of power measurements
   a. Keep track of the known data values d_i as recording the measurements
   b. For each d_i there exists a power trace of size T: $\mathbf{t'}_i = (t_{i,1}, \ldots, t_{i,T})$
3. Calculate hypothetical intermediate values
   a. For each k, the K possible choices are $\mathbf{k} = (k_1, \ldots, k_K)$
   b. The possible choices are used in conjunction with $f(d,k)$
4. Map hypothetical intermediate values to resulting predicted power values
5. Compare predicted power values with the actual trace values

Recall Slide 6 of Lecture 29 Power Analysis Part V: Microcontroller Example

- AES software SBOX calculation
  - s = S(*p* XOR *k*) where *p* is the plaintext and *k* is the subkey (each of size 8 bits)
- Calculate 1000 power traces
  - Each power trace corresponds to a specific plaintext input
    - Hence, there are 1000 power measurements taken for each value of the 128 bit plaintext
      - Note that the overall plaintext input size is 128 for AES, hence there are 2^{128} possible values of the overall plaintext
    - E.g., for a timeframe of 100 μs, there is a measurement for every 100 ns (10 MHz sample rate of the power measurement device, e.g., an oscilloscope)
  - A total of 1,000,000 measurements are stored in this example, e.g., from an oscilloscope based power measurement setup
    - If each measurement requires 32 bits (a word), then the filesize is 4 MB (Megabytes)

Recall Slide 17 of Lecture 28 Power Analysis Part V:

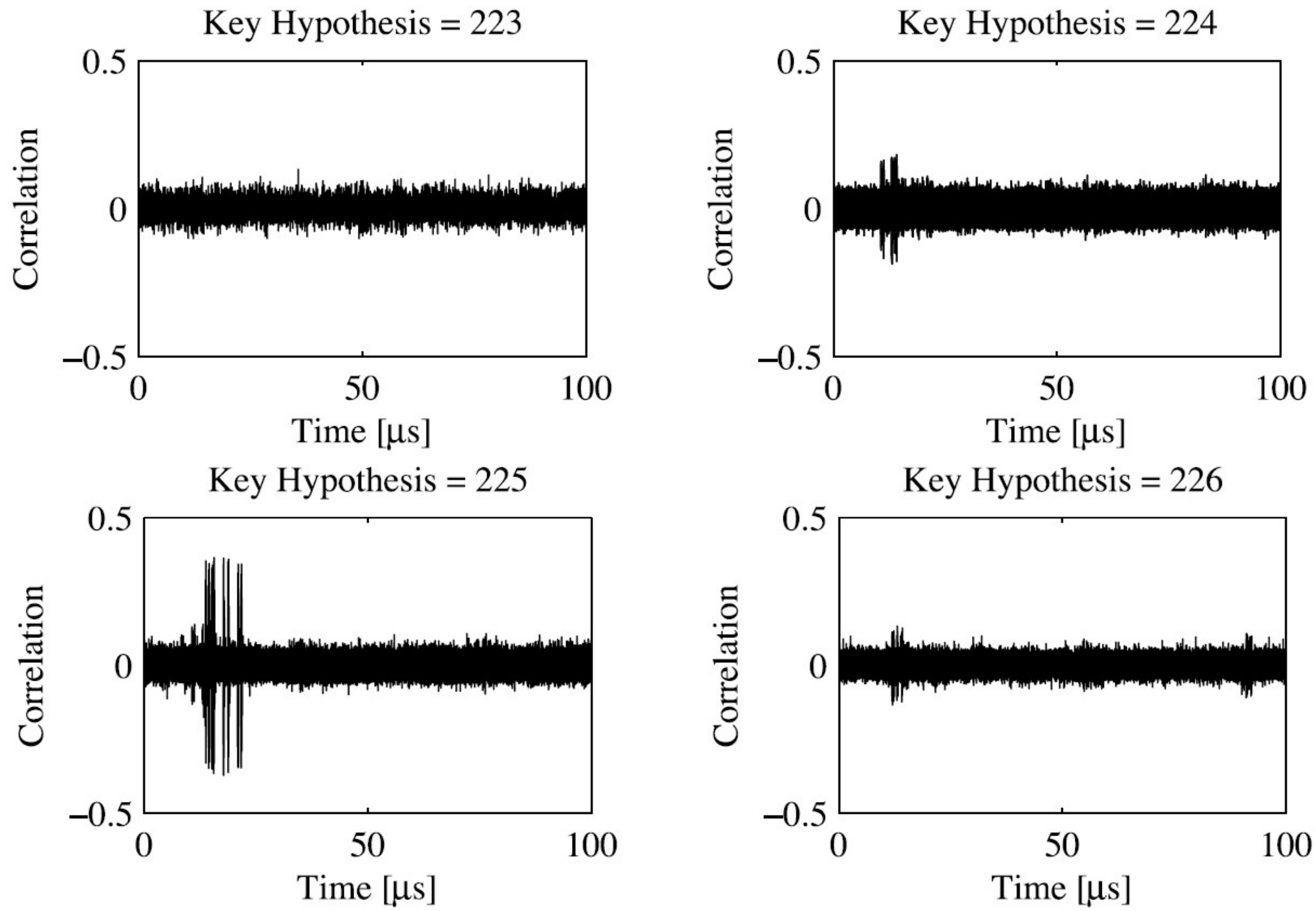
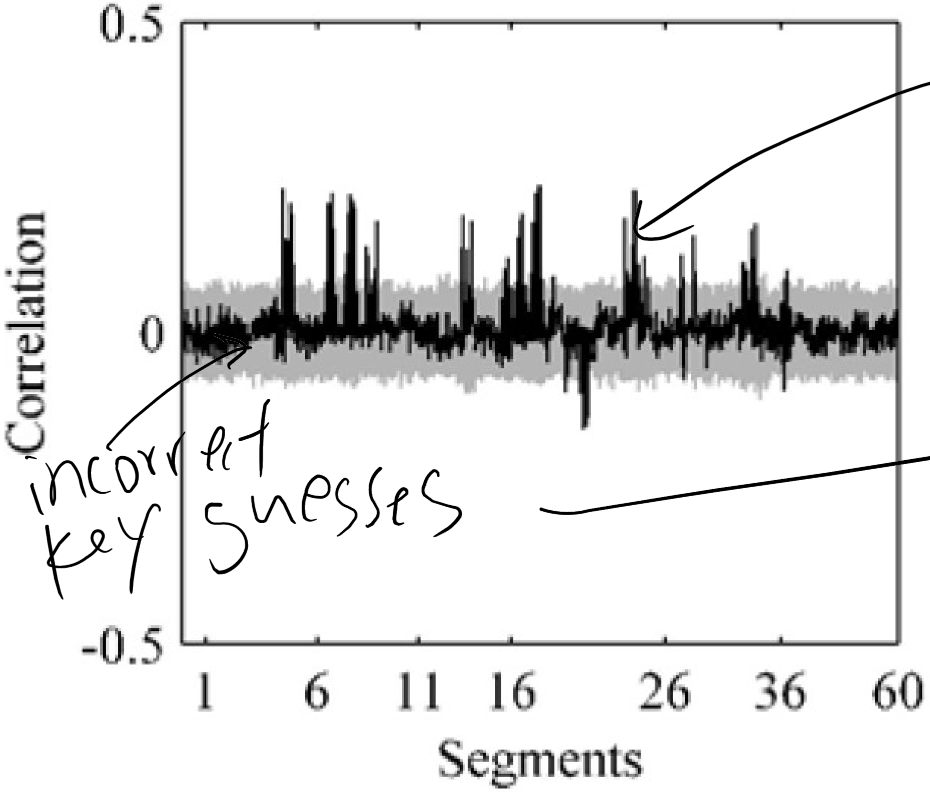


Figure 6.2. The rows of the matrix $\mathbf{R}$ that correspond to the key hypotheses 223, 224, 225, and 226.

Now Back to 2nd Order DPA…

2^{nd} (and Higher!) Order DPA Attacks

1. Choose an intermediate part of the algorithm to attack
   a. For example, function *f(d,k)* where *d* is a data input and *k* is a small part of the secret key stored in the device under attack
   b. Typically *d* is either plaintext or ciphertext
2. Make a large number of power measurements
   a. Keep track of the known data values d_i as recording the measurements
   b. For each d_i there exists a power trace of size T: $\mathbf{t'_i} = (t_{i,1}, \ldots, t_{i,T})$
   c. Preprocess the power traces; for 2^{nd} order, a window of size ℓ adds ~ℓ^2 preprocessing calculations
3. Calculate hypothetical intermediate values
   a. For each k, the K possible choices are $\mathbf{k} = (k_1, \ldots, k_K)$
   b. The possible choices are used in conjunction with *f(d,k)*
4. Map hypothetical intermediate values to resulting predicted power values
   a. For 2^{nd} order, duplicate the ~ℓ^2 preprocessing calculations
5. Compare predicted power values with the actual trace values
   a. Statistical tests involve differences, whether difference of means, covariance, or other



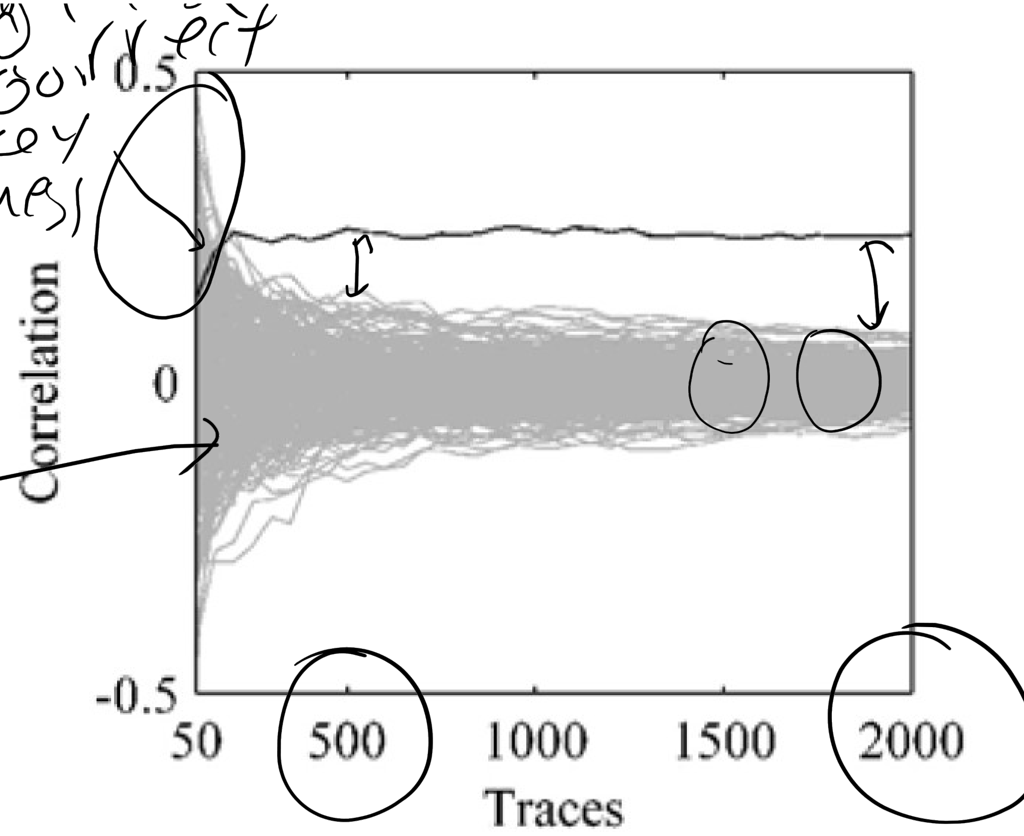


Figure 10.1. Result of a second-order DPA attack on a masked AES implementation in software.

Figure 10.2. Evolution of the correlation coefficient over an increasing number of traces.