Power Analysis Part V: DPA Cont'd

Cryptographic Hardware for Embedded Systems

ECE 3170

Fall 2026

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

©Georgia Institute of Technology, 2018-2026

Reading

- This lecture covers a portion of Differential Power Analysis as explained in Chapter 6 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.

©Georgia Institute of Technology, 2018-2026

Questions Answered by This Lecture

- How can one carry out DPA on a microcontroller running AES with the goal of finding the key?
  - Assumption # 1: the attacker can input any plaintext desired and obtain the corresponding ciphertext
  - Assumption # 2: the attacker can accurately measure power (either through physical access or remote access to the power trace data measurement)

©Georgia Institute of Technology, 2018-2026

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: $\boldsymbol{t'_i} = (t_{i,1}, \ldots, t_{i,T})$
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

©Georgia Institute of Technology, 2018-2026

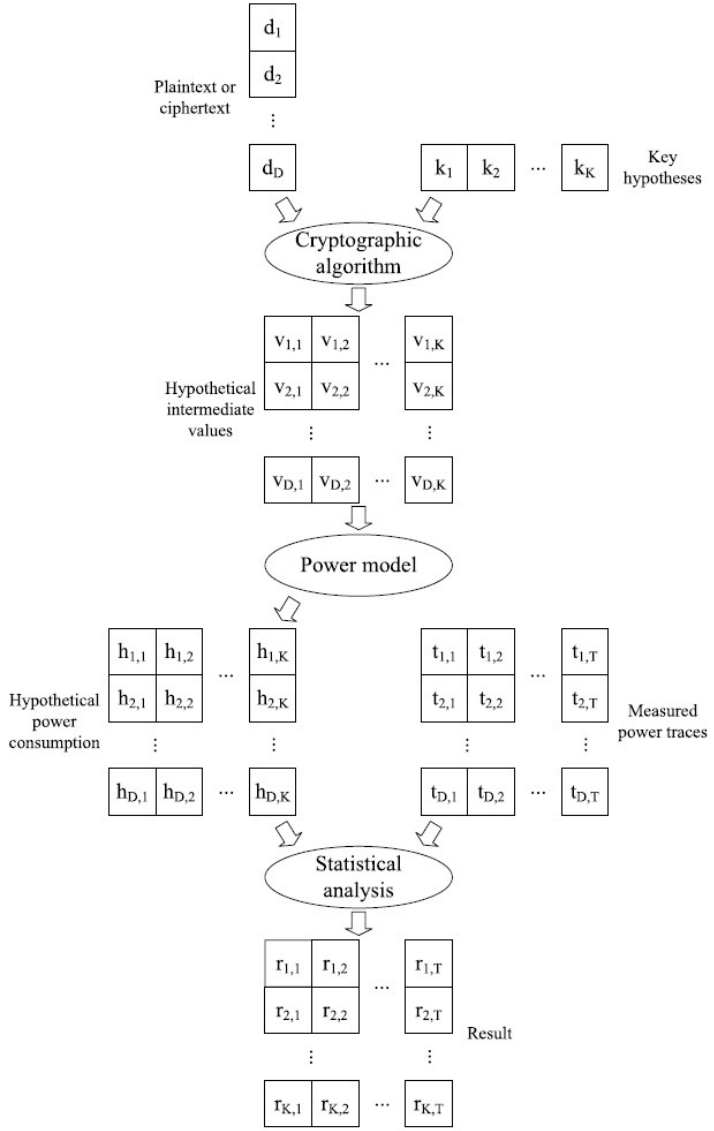
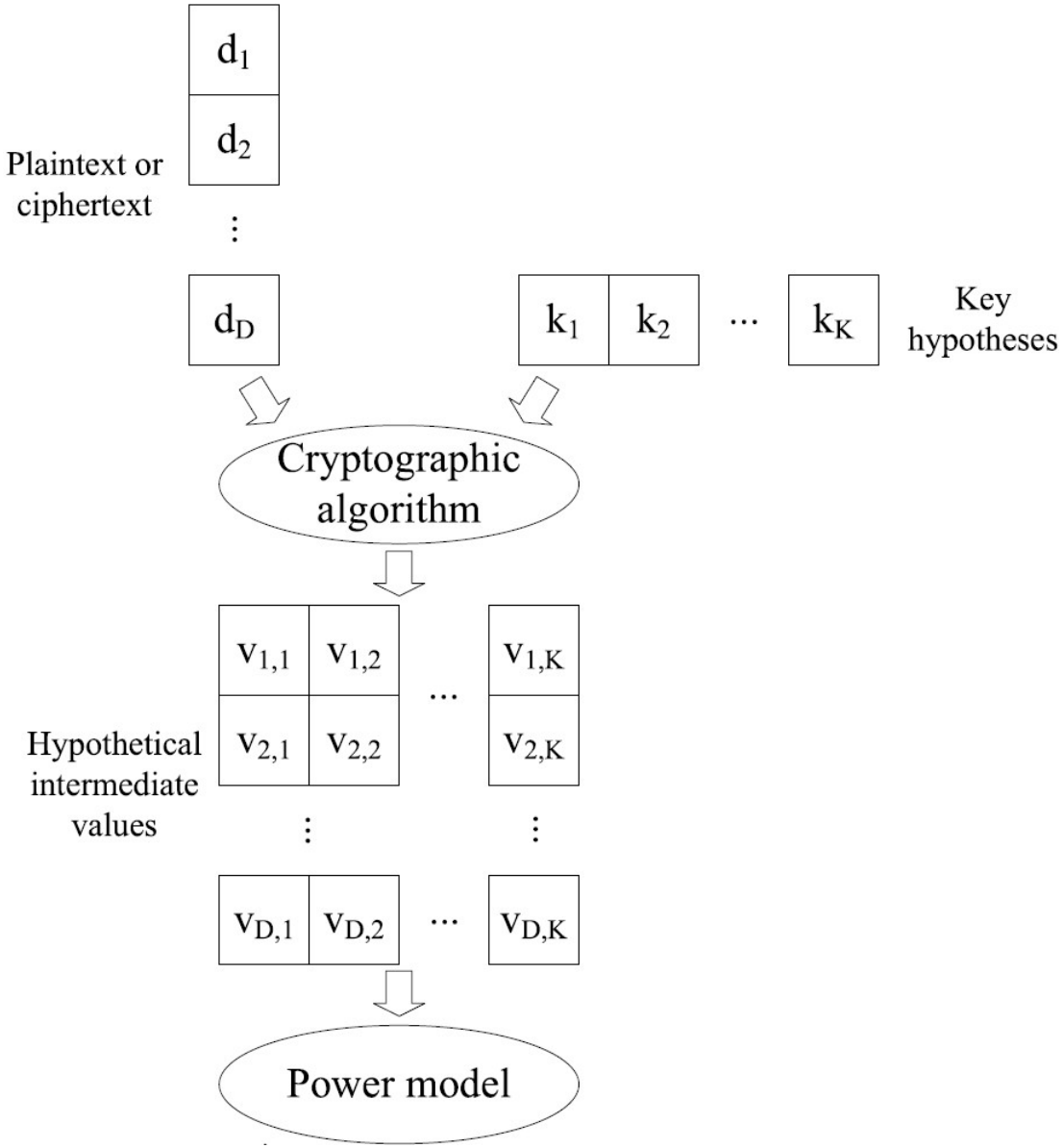


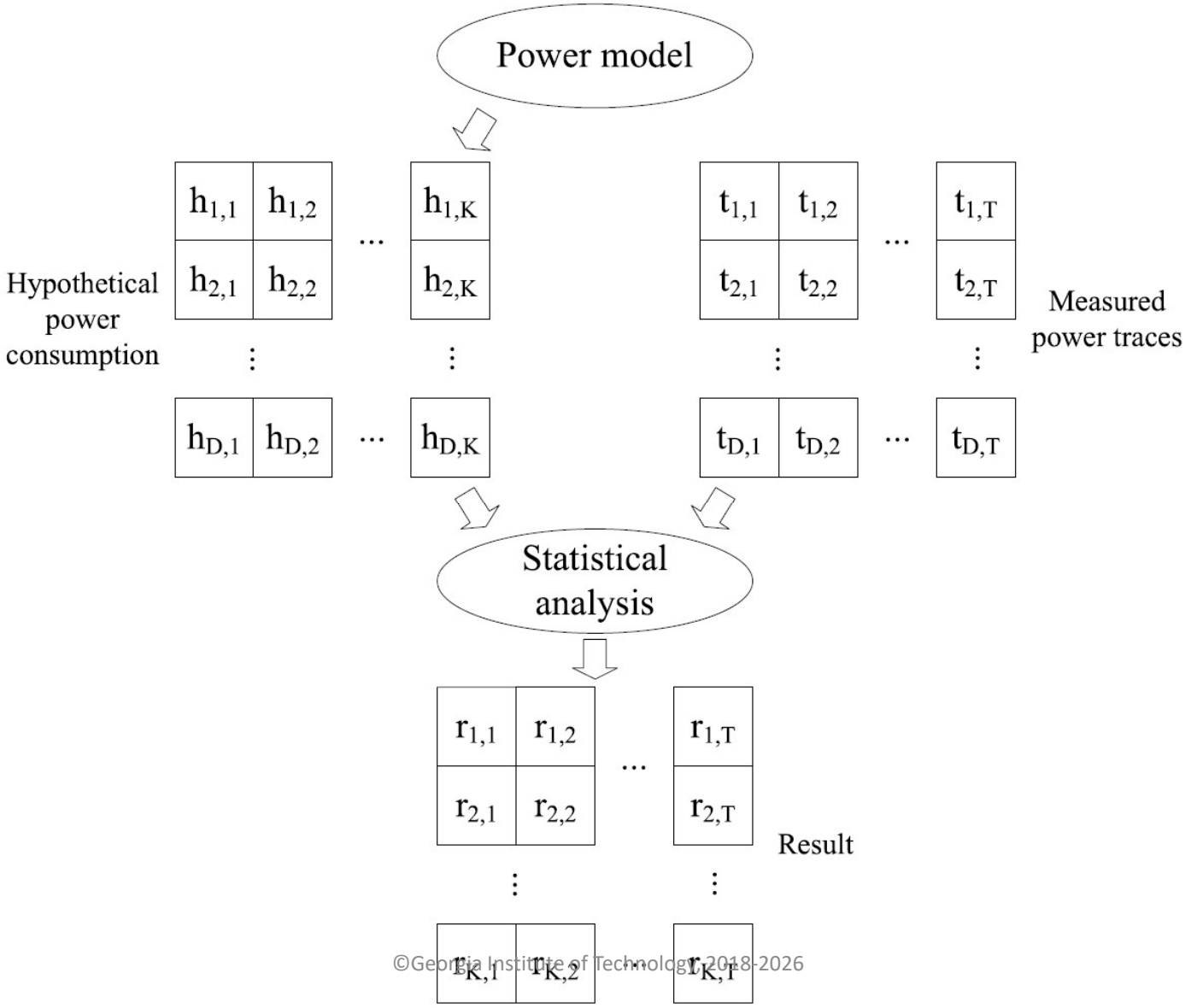
Figure 6.1. Block diagram illustrating the steps 3 to 5 of a DPA attack.

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)

©Georgia Institute of Technology, 2018-2026


Plaintext or ciphertext
d_1
d_2
d_D
k_1
k_2
k_K
Key hypotheses
Cryptographic algorithm
Hypothetical intermediate values
$v_{1,1}$ $v_{1,2}$ $v_{1,K}$
$v_{2,1}$ $v_{2,2}$ $v_{2,K}$
$v_{D,1}$ $v_{D,2}$ $v_{D,K}$
Power model



©Georgia Institute of Technology, 2018-2026

Microcontroller Example Continued

- AES software SBOX calculation
  - Let us consider the first byte (8 bits) of the overall plaintext input (128 bits)
  - s = S(*p* XOR *k*) where *p* is the plaintext and *k* is the subkey (each of size 8 bits)
- Recall we have 1000 power traces, each with a known specific value of *p*

©Georgia Institute of Technology, 2018-2026

Steps in Differential Power Analysis

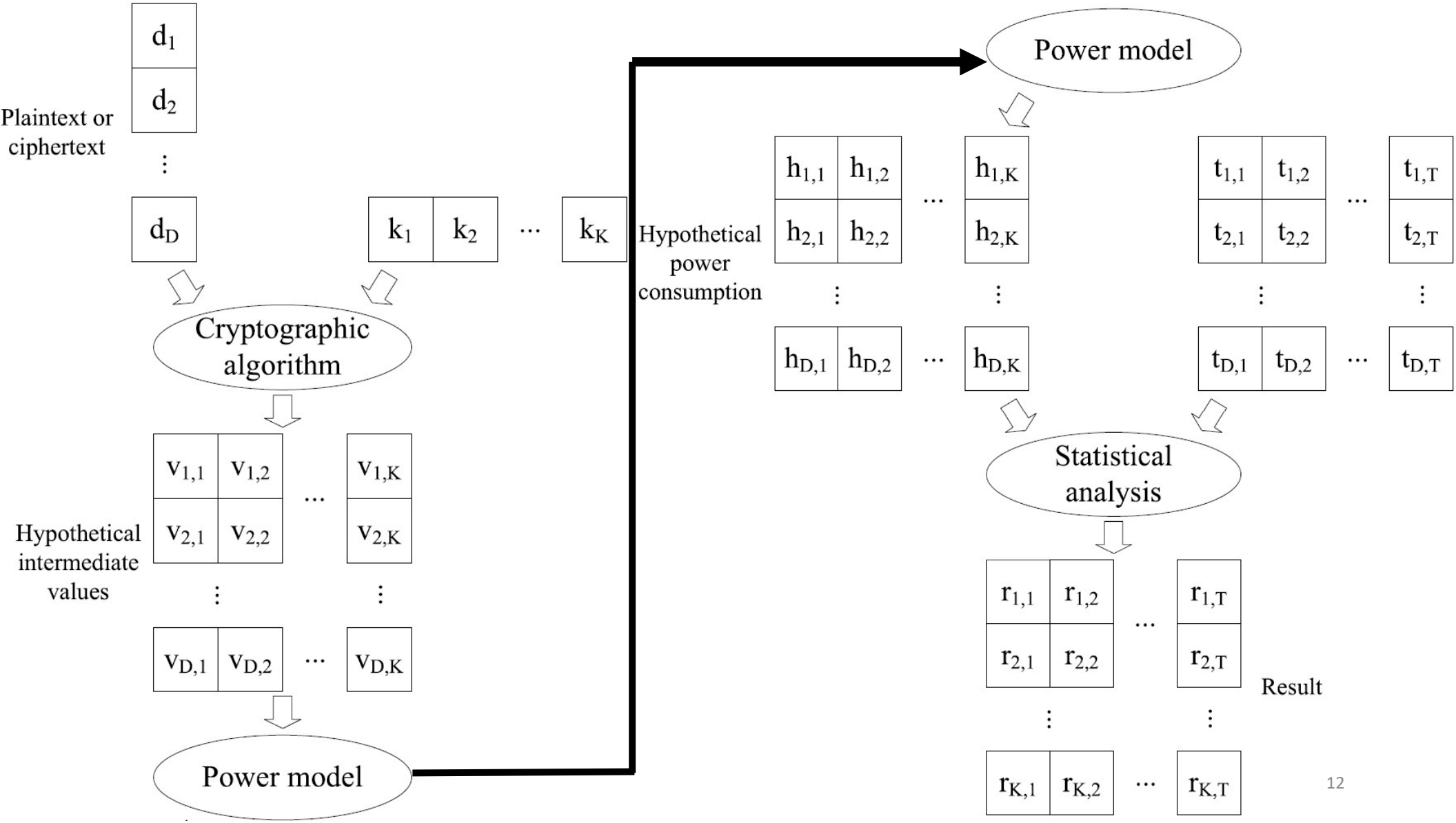
1. Choose an intermediate part of the algorithm to attack
   a. For example, function *f(d,k)* where *d* is a data input and *k* is a small part of the secret key stored in the device under attack
   b. Typically *d* is either plaintext or ciphertext
2. Make a large number of power measurements
   a. 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: $\boldsymbol{t'_i} = (t_{i,1}, \ldots, t_{i,T})$
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

©Georgia Institute of Technology, 2018-2026

Microcontroller Example Continued

- AES software SBOX calculation
  - Let us consider the first byte (8 bits) of the overall plaintext input (128 bits)
  - s = S(*p* XOR *k*) where *p* is the plaintext and *k* is the subkey (each of size 8 bits)
- Recall we have 1000 power traces, each with a known specific value of *p*
- Step 3 is to calculate hypothesized intermediate values of s = S(*p* XOR *k*)
  - Let $v_{i,j} = s_{i,j} = S(p_i \text{ XOR } k_j)$ where *i* ranges from 1 to 1000 and *j* from 0 to 256
  - Matrix **V** = $\{v_{i,j}\}$ has size 1000 rows by 256 columns

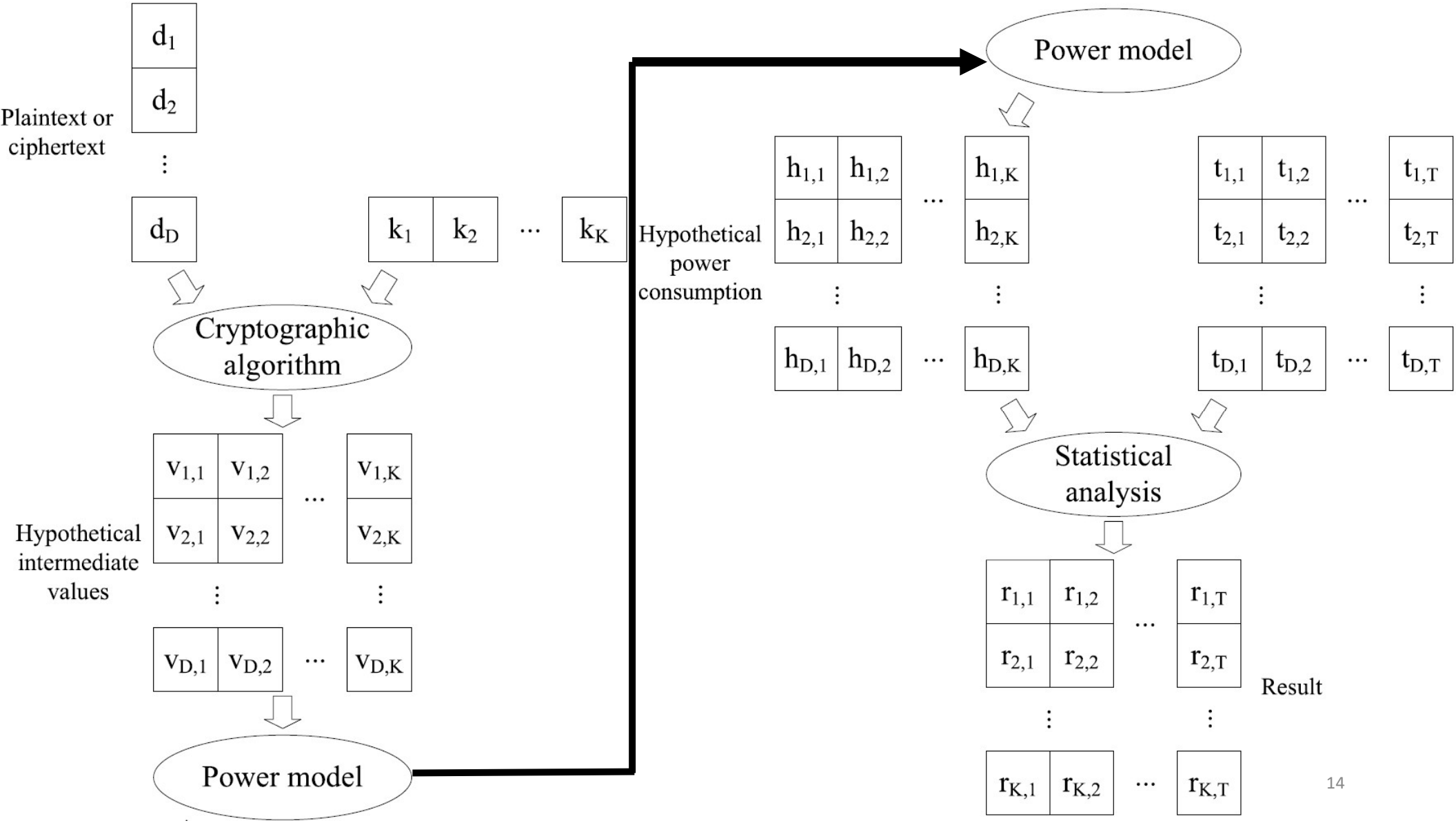
©Georgia Institute of Technology, 2018-2026


Plaintext or ciphertext
d_1
d_2
d_D
k_1 k_2 ... k_K
Cryptographic algorithm
Hypothetical intermediate values
$v_{1,1}$ $v_{1,2}$ ... $v_{1,K}$
$v_{2,1}$ $v_{2,2}$ $v_{2,K}$
$v_{D,1}$ $v_{D,2}$ ... $v_{D,K}$
Power model
Hypothetical power consumption
Power model
$h_{1,1}$ $h_{1,2}$ ... $h_{1,K}$
$h_{2,1}$ $h_{2,2}$ $h_{2,K}$
$h_{D,1}$ $h_{D,2}$ ... $h_{D,K}$
$t_{1,1}$ $t_{1,2}$ ... $t_{1,T}$
$t_{2,1}$ $t_{2,2}$ $t_{2,T}$
$t_{D,1}$ $t_{D,2}$ ... $t_{D,T}$
Statistical analysis
$r_{1,1}$ $r_{1,2}$ ... $r_{1,T}$
$r_{2,1}$ $r_{2,2}$ $r_{2,T}$
$r_{K,1}$ $r_{K,2}$ ... $r_{K,T}$
Result

Microcontroller Example Continued Again

- AES software SBOX calculation s = S(*p* XOR *k*)
- We have 1000 power traces each with a known specific value of *p*
- Step 3 is has calculated hypothesized intermediate values
  - **V** = $\{v_{i,j}\} = \{s_{i,j}\} = \{ S(p_i \text{ XOR } k_j) \}$ of size 1000 rows by 256 columns
- Step 4 in this example maps **V** to **H**
  - Use $h_{i,j} = LSB(v_{i,j}) = LSB(s_{i,j}) = LSB(S(p_i \text{ XOR } k_j))$
  - **H** = $\{h_{i,j}\}$ = results of an energy consumption model = an estimated set of power measurements based on a power model and assumed key values = a vector of size 1000 rows by 256 columns

©Georgia Institute of Technology, 2018-2026


Plaintext or ciphertext
d_1
d_2
d_D
k_1 k_2 ... k_K
Cryptographic algorithm
Hypothetical intermediate values
$v_{1,1}$ $v_{1,2}$ ... $v_{1,K}$
$v_{2,1}$ $v_{2,2}$ $v_{2,K}$
$v_{D,1}$ $v_{D,2}$ ... $v_{D,K}$
Power model
Hypothetical power consumption
Power model
$h_{1,1}$ $h_{1,2}$ ... $h_{1,K}$
$h_{2,1}$ $h_{2,2}$ $h_{2,K}$
$h_{D,1}$ $h_{D,2}$ ... $h_{D,K}$
$t_{1,1}$ $t_{1,2}$ ... $t_{1,T}$
$t_{2,1}$ $t_{2,2}$ $t_{2,T}$
$t_{D,1}$ $t_{D,2}$ ... $t_{D,T}$
Statistical analysis
$r_{1,1}$ $r_{1,2}$ ... $r_{1,T}$
$r_{2,1}$ $r_{2,2}$ $r_{2,T}$
$r_{K,1}$ $r_{K,2}$ ... $r_{K,T}$
Result

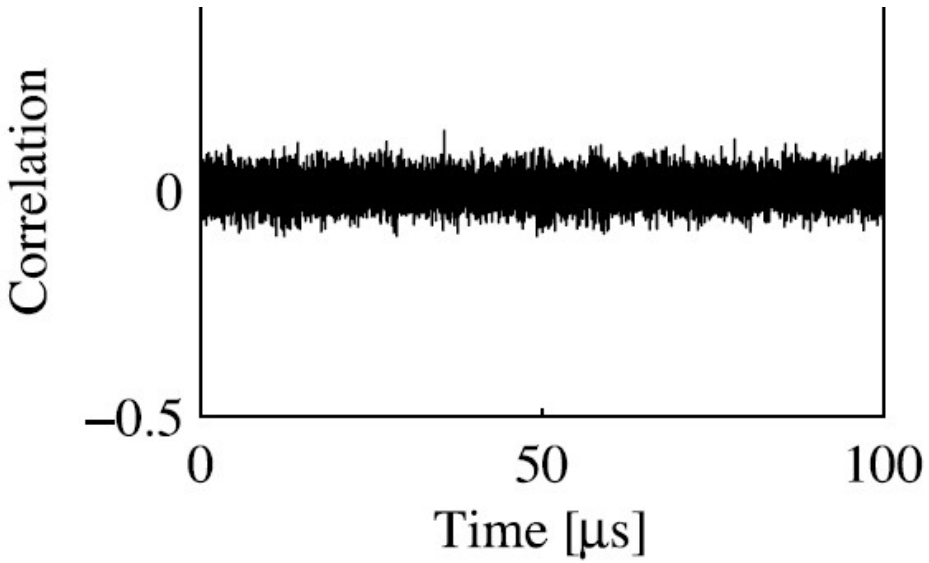
Microcontroller Example Continued Again

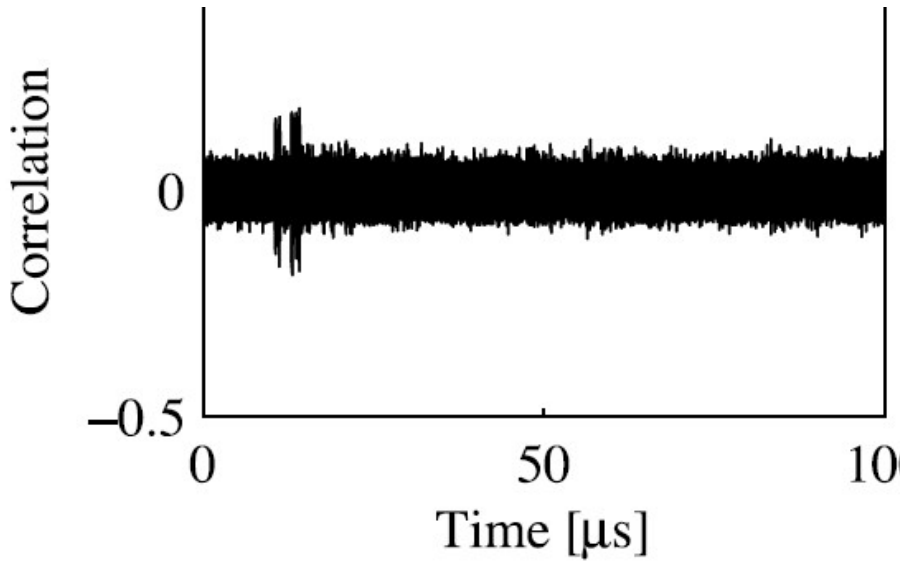
- AES software SBOX calculation s = S(*p* XOR *k*)
- We have 1000 power traces each with a known specific value of *p*
- Step 3 is has calculated hypothesized intermediate values
  - **V** = $\{v_{i,j}\} = \{s_{i,j}\} = \{ S(p_j \text{ XOR } k_j) \}$ of size 1000 rows by 256 columns
- Step 4 in this example maps **V** to **H**
  - Use $h_{i,j} = LSB(v_{i,j}) = LSB(s_{i,j}) = LSB(S(p_j \text{ XOR } k_j))$
  - **H** = $\{h_{i,j}\}$ = results of an energy consumption model = an estimated set of power measurements based on a power model and assumed key values = a vector of size 1000 rows by 256 columns
- Step 5 is to calculate the covariance of **H** with the actual trace data

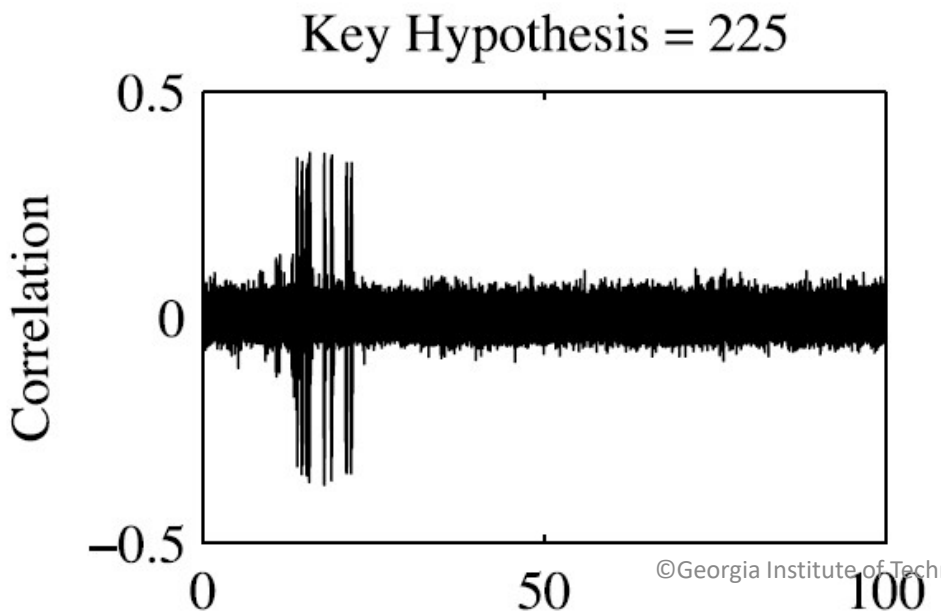
©Georgia Institute of Technology, 2018-2026

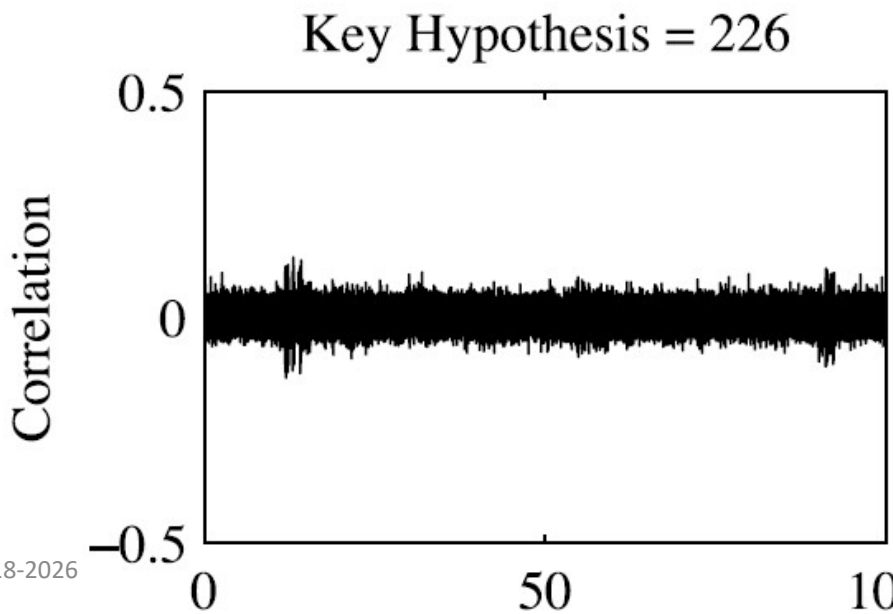
$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i) \cdot (t_{d,j} - \bar{t}_j)}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i)^2 \cdot \sum_{d=1}^{D}(t_{d,j} - \bar{t}_j)^2}} \tag{6.2}$$

©Georgia Institute of Technology, 2018-2026









©Georgia Institute of Technology, 2018-2026

Comments

©Georgia Institute of Technology, 2018-2026

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., $-1 \leq \rho \leq 1$ and also thus $-1 \leq r \leq 1$

©Georgia Institute of Technology, 2018-2026

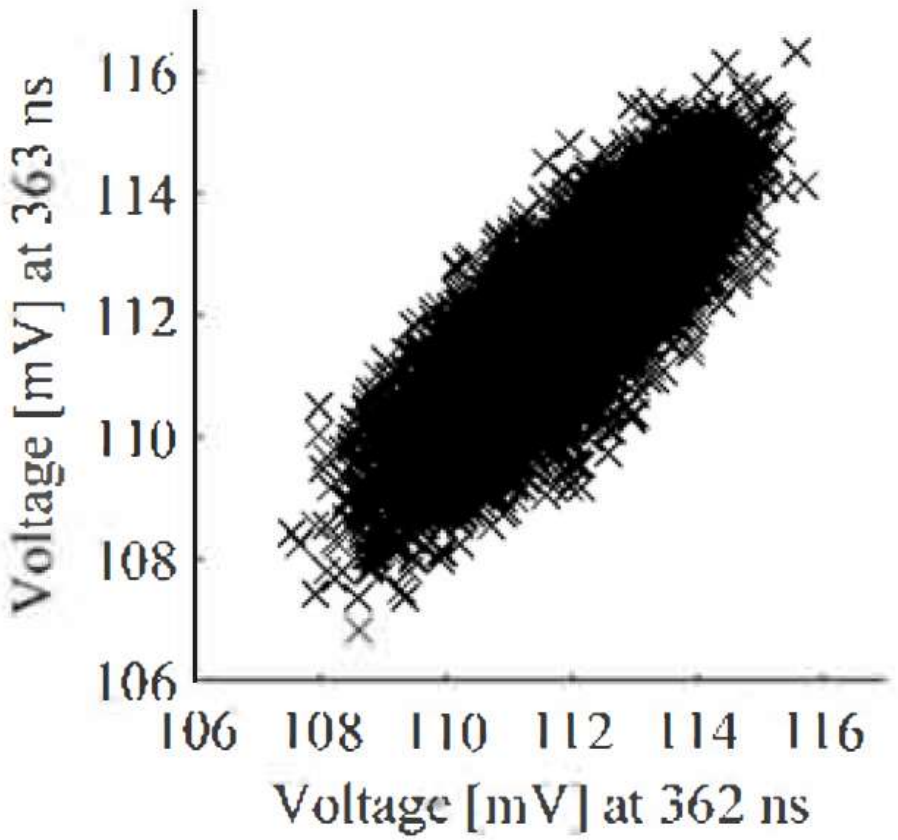


Figure 4.9. Scatter Plot: The power consumption at 362 ns is correlated to the power consumption at 363 ns. $r = 0.82$

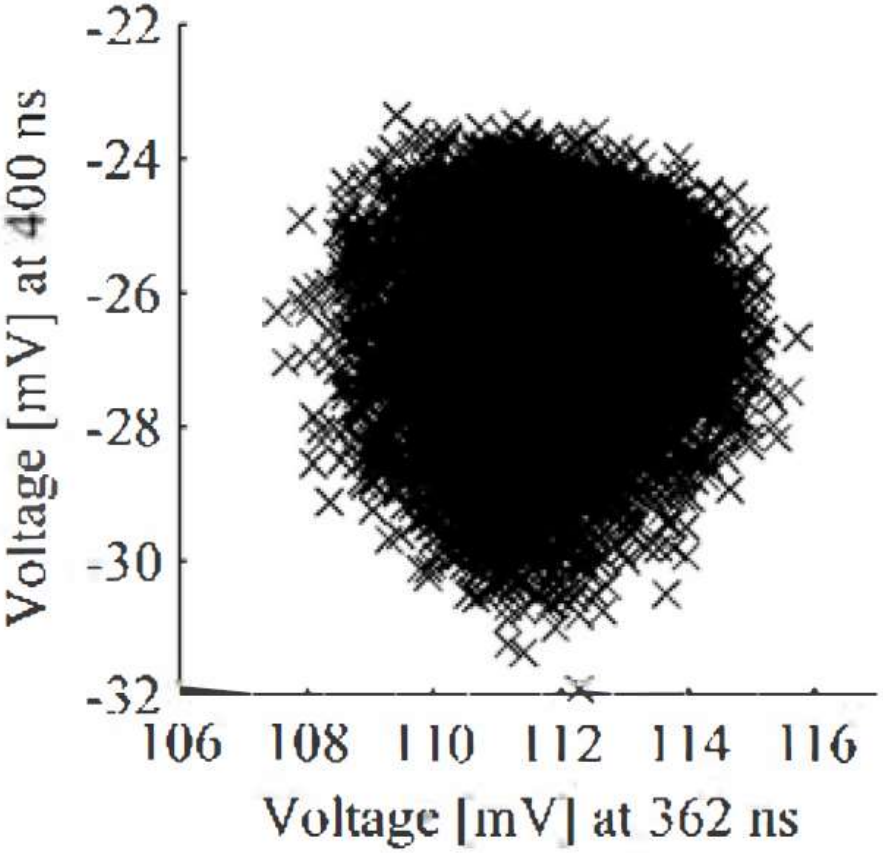


Figure 4.10. Scatter Plot: The power consumption at 362 ns is largely uncorrelated to the power consumption at 400 ns. $r = 0.12$

©Georgia Institute of Technology, 2018-2026