# NPTEL ONLINE CERTIFICATION COURSES

## Course Name: Hardware Security

**Faculty Name: Prof Debdeep Mukhopadhyay**

**Department : Computer Science and Engineering**

## Topic

**Lecture 31: Power Analysis-VII**

## CONCEPTS COVERED

Concepts Covered:

❑Correlation Power Attacks (CPA)

❑Implementing CPA on Simulated Traces

# Correlation Power Attack (CPA)

- Like DoM based DPA, CPA also relies on targeting an intermediate computation, typically the input or output of an S-Box.
- These input values are computed from a known value, say the ciphertext and a portion of the key, which is guessed.
- The power model is then subsequently applied to develop a hypothetical power trace of the device for a given input to the cipher.
- These hypothetical power values are then stored in a matrix for several inputs and can be indexed by the known value of the ciphertext and the guessed key byte.
- This matrix is denoted as H, the hypothetical power matrix.

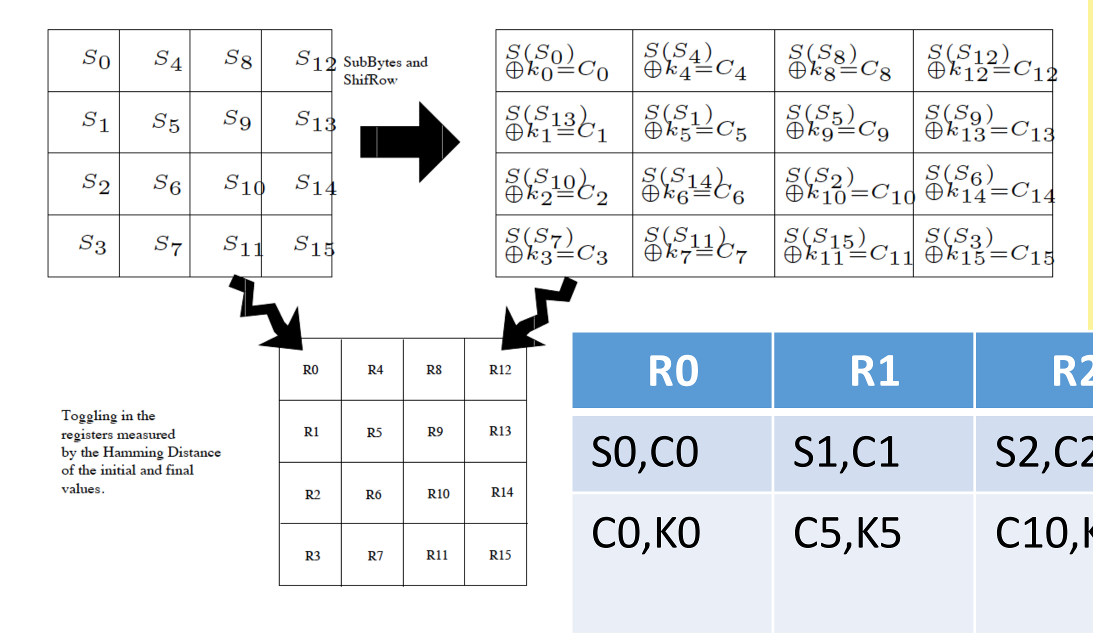
# Correlation Power Attack (CPA)-Contd.

- The attacker also observes the actual power traces, and stores them in a matrix for several inputs.
- The actual power values can be indexed by the known value of the ciphertext and the time instance when the power value was observed.
- This matrix is denoted by T, the real power matrix.
- It may be observed that one of the columns of the matrix H corresponds to the correct key k*.
- CPA tries to compute the **similarity** between the columns of the matrix H and the columns of the matrix T, to distinguish k* from rest: similarity computed by Pearson's Correlation, usually.

# Computing Correlation Coefficient for Simulated Power Traces for AES

- Like before, we simulate the power profile for the iterative AES, this time by using Hamming Distance power model applied on the state registers updated after each round.
- The real power matrix is stored in the array trace[NSample][NPoint]
  - NSample: Number of Power Traces acquired
  - NPoint: Time instances for which the power values are observed. Here NPoint is 12.

# Calculating the Hypothetical Power



| | | | |
|---|---|---|---|
| $S_0$ | $S_4$ | $S_8$ | $S_{12}$ |
| $S_1$ | $S_5$ | $S_9$ | $S_{13}$ |
| $S_2$ | $S_6$ | $S_{10}$ | $S_{14}$ |
| $S_3$ | $S_7$ | $S_{11}$ | $S_{15}$ |

SubBytes and ShifRow

| | | | |
|---|---|---|---|
| $S(S_0) \oplus k_0 = C_0$ | $S(S_4) \oplus k_4 = C_4$ | $S(S_8) \oplus k_8 = C_8$ | $S(S_{12}) \oplus k_{12} = C_{12}$ |
| $S(S_{13}) \oplus k_1 = C_1$ | $S(S_1) \oplus k_5 = C_5$ | $S(S_5) \oplus k_9 = C_9$ | $S(S_9) \oplus k_{13} = C_{13}$ |
| $S(S_{10}) \oplus k_2 = C_2$ | $S(S_{14}) \oplus k_6 = C_6$ | $S(S_2) \oplus k_{10} = C_{10}$ | $S(S_6) \oplus k_{14} = C_{14}$ |
| $S(S_7) \oplus k_3 = C_3$ | $S(S_{11}) \oplus k_7 = C_7$ | $S(S_{15}) \oplus k_{11} = C_{11}$ | $S(S_3) \oplus k_{15} = C_{15}$ |

Toggling in the registers measured by the Hamming Distance of the initial and final values.

| | | | |
|---|---|---|---|
| R0 | R4 | R8 | R12 |
| R1 | R5 | R9 | R13 |
| R2 | R6 | R10 | R14 |
| R3 | R7 | R11 | R15 |

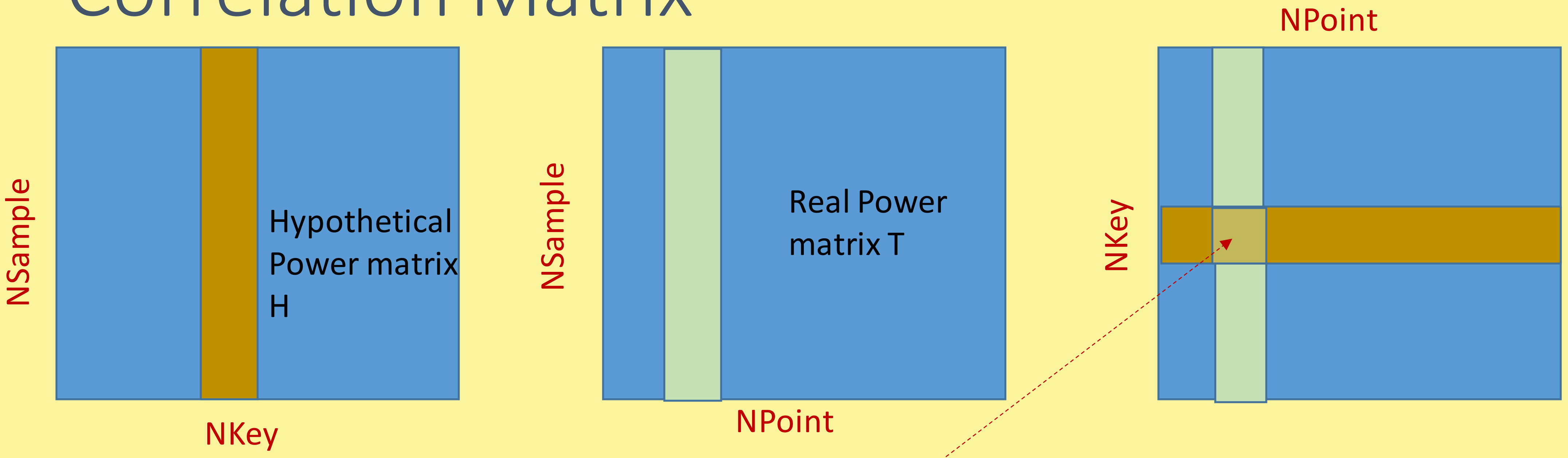
| R0 | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| S0,C0 | S1,C1 | S2,C2 | S3,C3 | S4,C4 | S5,C5 | S6,C6 | S7,C7 |
| C0,K0 | C5,K5 | C10,K10 | C15,K15 | C4,K4 | C9,K9 | C14,K14 | C3,K3 |

| R8 | R9 | R10 | R11 | R12 | R13 | R14 | R15 |
|---|---|---|---|---|---|---|---|
| S8,C8 | S9,C9 | S10,C10 | S11,C11 | S12,C12 | S13,C13 | S14,C14 | S15,C15 |
| C8,K8 | C13,K13 | C2,K2 | C7,K7 | C12,K12 | C1,K1 | C6,K6 | C11,K11 |

# Computing the Correlation Matrix

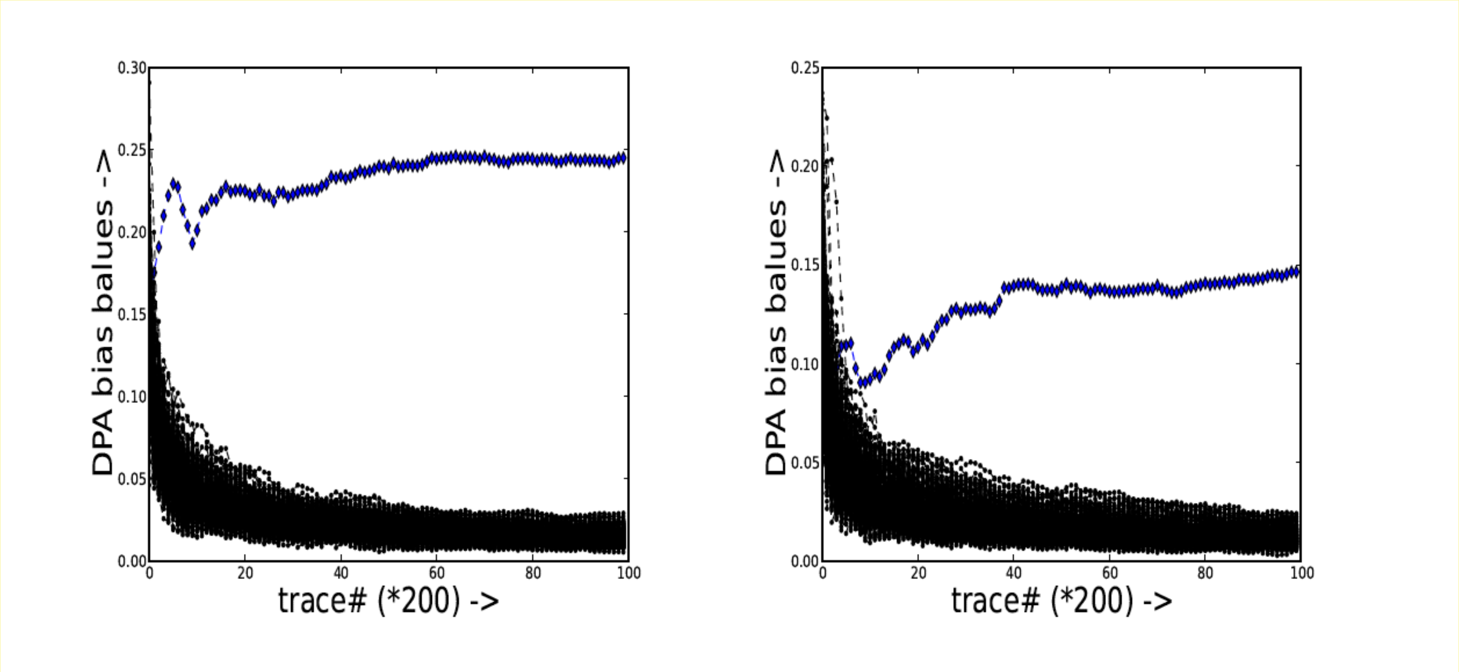
- Actual Power values for all the NSample encryptions are stored in the array trace[NSample][NPoint].
  - However, reflect as we are calculating hypothetical power using HD, the trick which we applied before for storing the traces compactly will not work.
  - Attacker first scans each column of this array and computes the average, and stores in meanTrace[NPoint].
- Likewise, the hypothetical power is stored in the array hPower[NSample][NKey].
  - Attacker scans each column and stores in the array meanH[NKey]

# Correlation Matrix



$$C[i][j] = \frac{\Sigma_{k=0}^{NSample}(hPower[i][k] - meanH[i])(trace[j][k] - meanTrace[j])}{\Sigma_{k=0}^{NSample}(hPower[i][k] - meanH[i])^2 \Sigma_{k=0}^{NSample}(trace[j][k] - meanTrace[j])^2}$$

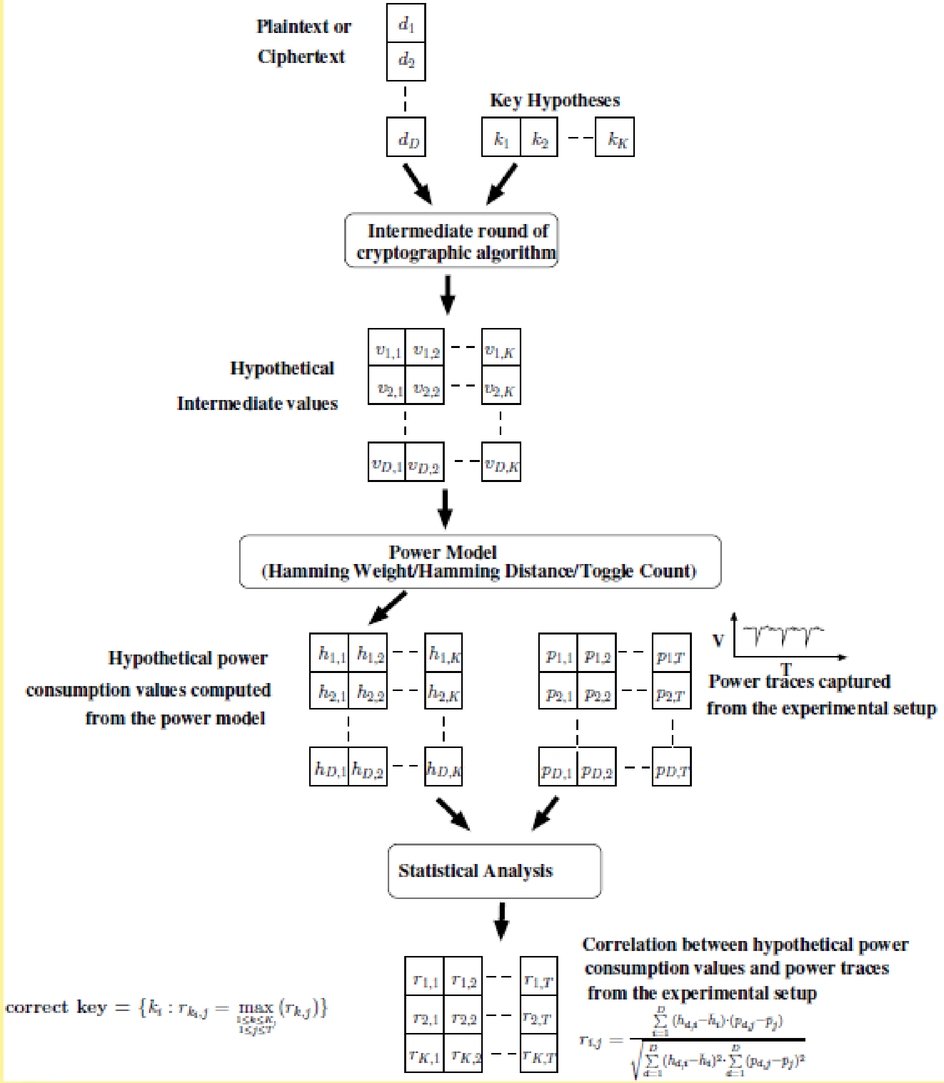
# Experimental Results on Simulated Traces



CPA on first key byte of 10th Round of AES
Power is Simulated by Hamming Distance of the Registers after each Round

Power is Simulated by Hamming Distance of the Registers after each Round, with superimposed Gaussian Noise.

# Summary of Correlation Power Analysis



Plaintext or Ciphertext

Key Hypotheses

Intermediate round of cryptographic algorithm

Hypothetical Intermediate values

Power Model (Hamming Weight/Hamming Distance/Toggle Count)

Hypothetical power consumption values computed from the power model

Power traces captured from the experimental setup

Statistical Analysis

Correlation between hypothetical power consumption values and power traces from the experimental setup

$$\text{correct key} = \{k_i : r_{k_i,j} = \max_{\substack{1\le k\le K,\\ 1\le j\le T}} (r_{k,j})\}$$

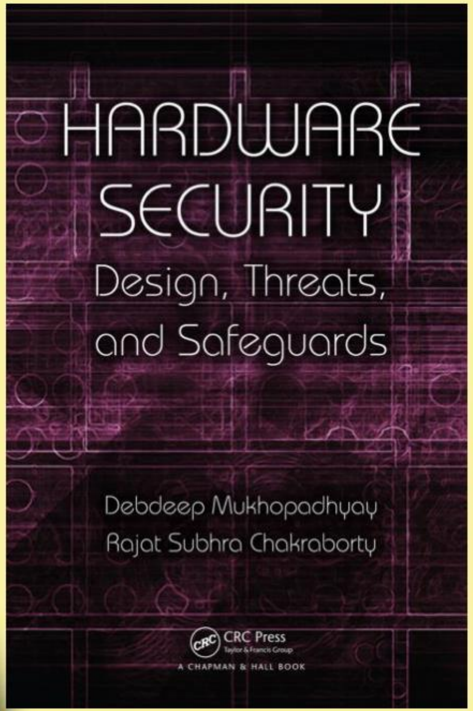
$$r_{i,j} = \frac{\sum_{d=1}^{D} (h_{d,i}-\bar{h}_i)\cdot(p_{d,j}-\bar{p}_j)}{\sqrt{\sum_{d=1}^{D} (h_{d,i}-\bar{h}_i)^2 \cdot \sum_{d=1}^{D} (p_{d,j}-\bar{p}_j)^2}}$$

$d_1$
$d_2$
$d_D$
$k_1$ $k_2$ -- $k_K$
$v_{1,1}$ $v_{1,2}$ -- $v_{1,K}$
$v_{2,1}$ $v_{2,2}$ -- $v_{2,K}$
$v_{D,1}$ $v_{D,2}$ -- $v_{D,K}$
$h_{1,1}$ $h_{1,2}$ -- $h_{1,K}$
$h_{2,1}$ $h_{2,2}$ -- $h_{2,K}$
$h_{D,1}$ $h_{D,2}$ -- $h_{D,K}$
$p_{1,1}$ $p_{1,2}$ -- $p_{1,T}$
$p_{2,1}$ $p_{2,2}$ -- $p_{2,T}$
$p_{D,1}$ $p_{D,2}$ -- $p_{D,T}$
V
T
$r_{1,1}$ $r_{1,2}$ -- $r_{1,T}$
$r_{2,1}$ $r_{2,2}$ -- $r_{2,T}$
$r_{K,1}$ $r_{K,2}$ -- $r_{K,T}$

# References

## References:

- ❑ Debdeep Mukhopadhyay and Rajat Subhra Chakraborty, Hardware Security: Design, Threats and Safeguards, CRC Press



D. Stinson, Cryptography: Theory and Practice, Chapman & Hall/CRC

Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography, Chapman & Hall/CRC

Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, An Introduction to Mathematical Cryptography, Springer.

## Conclusion:

**DoM based DPA can be programmed in a systematic fashion.**

**One can learn the technique on simulated power traces, modeled by Hamming Weight or Hamming Distance Models.**

**Stochastic Modeling of leakage can help to improve accuracy, but requires profiling.**

**CPA is an improved method of DPA and can also be systematically programmed.**

FREE ONLINE EDUCATION

swayam

शिक्षित भारत, उन्नत भारत

## NPTEL ONLINE CERTIFICATION COURSES

# Thank you!