

3: Power Analysis Attacks

Yunsi Fei

**Northeastern University
ECE Department**

DFA Example

- Assume ciphertext/faulty text below, assume the fault is on MI state

| | | | |
|----------------|----------------|-----------------|-----------------|
| C ₀ | C ₄ | C ₈ | C ₁₂ |
| C ₁ | C ₅ | C ₉ | C ₁₃ |
| C ₂ | C ₆ | C ₁₀ | C ₁₄ |
| C ₃ | C ₇ | C ₁₁ | C ₁₅ |

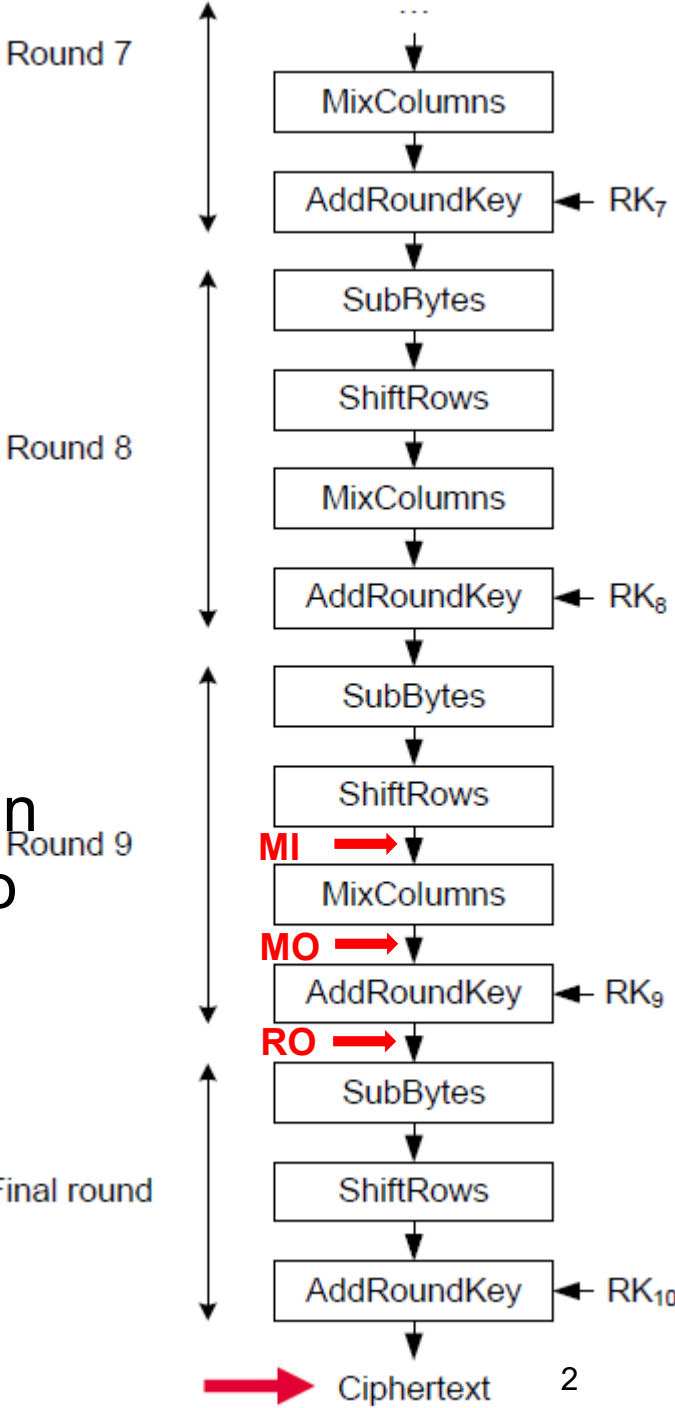
| | | | |
|----------------|----------------|-----------------|-----------------|
| F ₀ | F ₄ | F ₈ | F ₁₂ |
| F ₁ | F ₅ | F ₉ | F ₁₃ |
| F ₂ | F ₆ | F ₁₀ | F ₁₄ |
| F ₃ | F ₇ | F ₁₁ | F ₁₅ |

- The fault must fall on the second column of MI, and assume we know the row too

| | | | |
|-----------------|-------------------|------------------|------------------|
| MI ₀ | MI ₄ ' | MI ₈ | MI ₁₂ |
| MI ₁ | MI ₅ | MI ₉ | MI ₁₃ |
| MI ₂ | MI ₆ | MI ₁₀ | MI ₁₄ |
| MI ₃ | MI ₇ | MI ₁₁ | MI ₁₅ |

MI₄' = MI₄ ⊕ Δ

| | | | |
|-----------------|-------------------|------------------|------------------|
| MO ₀ | MO ₄ ' | MO ₈ | MO ₁₂ |
| MO ₁ | MO ₅ ' | MO ₉ | MO ₁₃ |
| MO ₂ | MO ₆ ' | MO ₁₀ | MO ₁₄ |
| MO ₃ | MO ₇ ' | MO ₁₁ | MO ₁₅ |



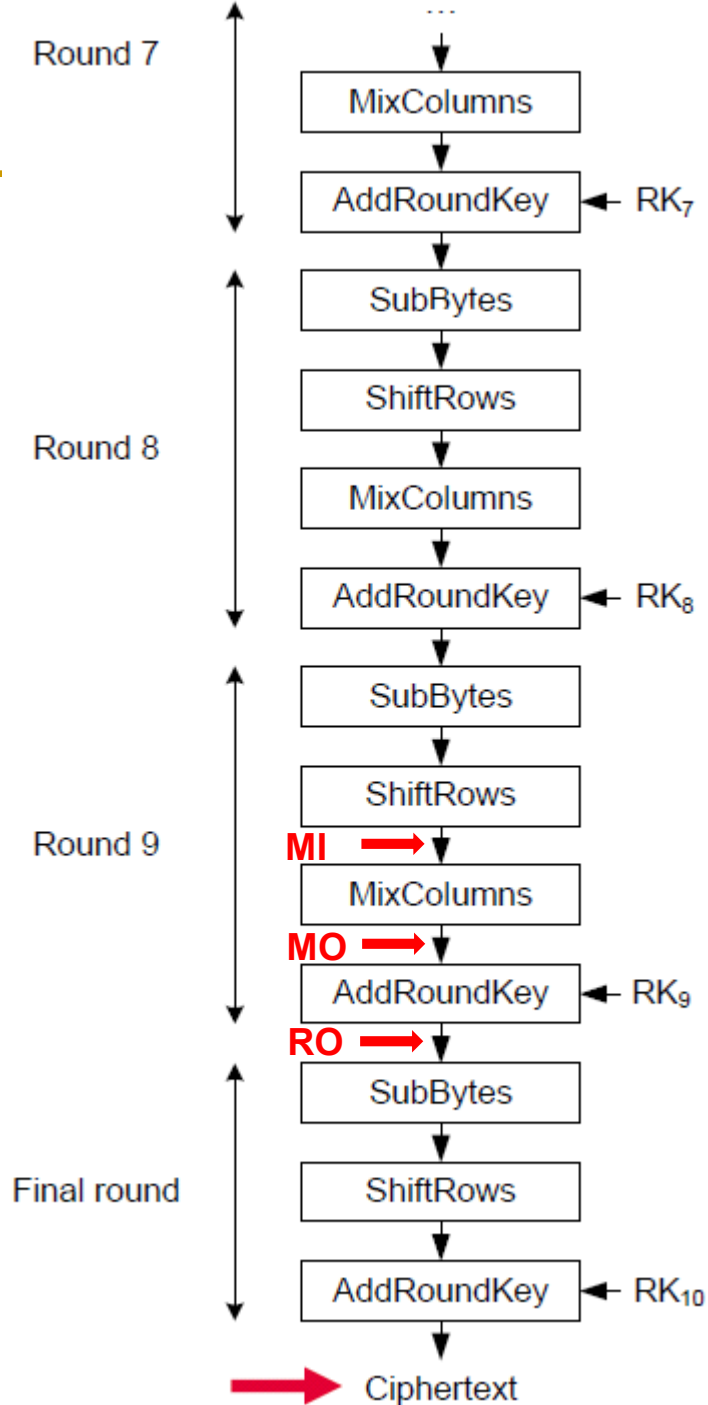
DFA Example

- The output state of round 9:

$$RO_i = S^{-1}(C_j \oplus K_j)$$

| | | | |
|--------|---------|-----------|-----------|
| RO_0 | RO_4' | RO_8 | RO_{12} |
| RO_1 | RO_5' | RO_9 | RO_{13} |
| RO_2 | RO_6' | RO_{10} | RO_{14} |
| RO_3 | RO_7' | RO_{11} | RO_{15} |

| | |
|-----|-----|
| i | j |
| 4 | 4 |
| 5 | 1 |
| 6 | 14 |
| 7 | 11 |

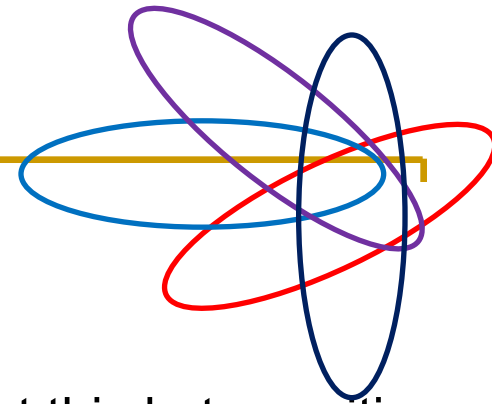


Algorithm of the Improved Attack

1. Calculate the list of 255 possible differentials at MO – DMO[255], each consists of four bytes ($DMO_4, DMO_5, DMO_6, DMO_7$) and corresponds to a fault value.
$$DMO[i] = \text{MixColumns}[i, 0, 0, 0], \quad i=1, \dots, 255$$
2. The differential propagates through AddRoundKey:
 $DRO[255]=DMO[255]$. These are the **predicted** differentials at RO state
3. For the first byte of the affected 4 bytes of the ciphertext
 - Calculate $\Delta RO_1 = S^{-1}(C_4 \oplus K_4) \oplus S^{-1}(F_4 \oplus K_4)$ for each possible value of K_4 , where S^{-1} is the inverse of Sbox, ΔRO_1 is an array of 256 bytes
$$\Delta RO_1[i] = S^{-1}(C_4 \oplus K_{4,i}) \oplus S^{-1}(F_4 \oplus K_{4,i}), \quad i=0, \dots, 255$$
 - Check whether this key leads to a valid differential at byte position 1 of DRO
 - Generate a list of potential key candidates and faults by storing
 - The key byte value
 - The index of the matching entry in DRO (1...255), i.e., fault Δ index

| K_4 | Δ index |
|-------|----------------|
| | |
| | |

Contd.



4. For each subsequent byte position do
- ❑ Calculate $\Delta RO_2, \Delta RO_3, \Delta RO_4$
 - ❑ Check whether this key leads to a valid differential at this byte position of DRO, and obtain another list

| K_1 | Δ index |
|-------|----------------|
| | |
| | |

| K_{14} | Δ index |
|----------|----------------|
| | |
| | |

| K_{11} | Δ index |
|----------|----------------|
| | |
| | |

5. Find the intersect of the four lists of Δ indices
6. The solutions in the intersect list may not be unique, go for another plaintext input with the same fault injection to obtain another set of $\{C, F\}$, repeat 2-5, until a unique fault is found and all the key byte values retrieved

Statistics and Complexity

- With one {C,F} pair
 - K_4 , fault space: 126
 - K_1 , fault space : 62
 - K_{14} , fault space: 31
 - K_{11} , fault space: 15
- With the second {C, F} pair
 - K_4 , fault space: 7
 - K_1 , fault space : 3
 - K_{14} , fault space: 3
 - K_{11} , fault space: 1
- The computational complexity: $4 \cdot 2^8$

Power Analysis Attacks

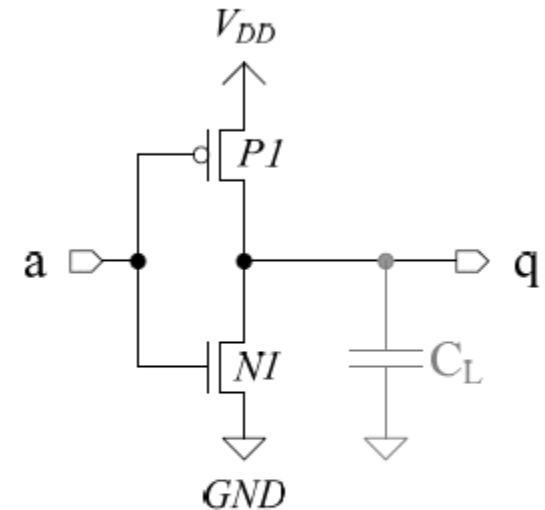
- The power consumption of cryptographic devices is typically the most critical side channel
- Power analysis attacks are very **powerful**: unless explicit countermeasures are implemented, every implementation can be broken
- Power analysis attacks are **cheap**: for an unprotected implementation all that is needed is a PC and a digital oscilloscope or A/D card (less than \$5000)

Overview

- Power consumption of CMOS circuits
- Basic analysis and processing techniques for side-channel information
- The classical DPA attack
- Countermeasures

Power Consumption Basics

- For power consumption to leak information, it has to be data-dependent
- Dynamic power consumption
 - Charging current: Whenever the output switches, the output capacitance needs to be charged or discharged; charging leads to higher current -> **the charging current is data-dependent**



A CMOS inverter
(the most simple gate)

Power Model

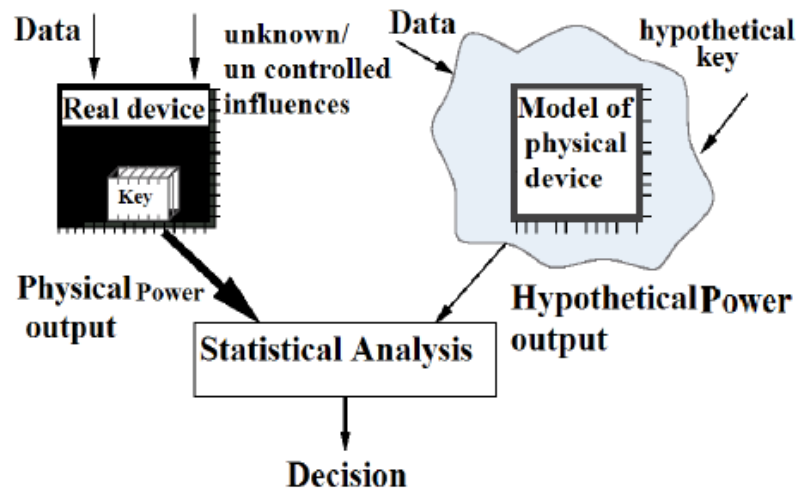
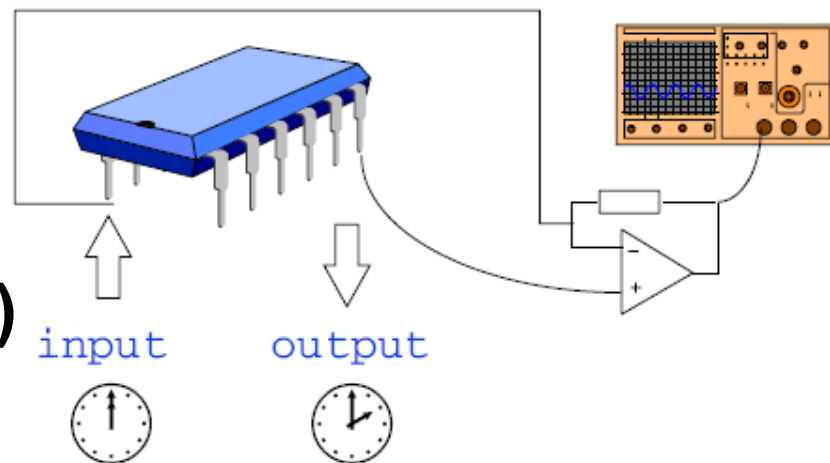
- Commonly used high-level models for the power consumption (p) of a circuit processing of a data sequence d_1, d_2, d_3, \dots
 - Hamming weight model: $p_i = HW(d_i)$, where HW calculates the Hamming weight (i.e., counts the number of 1s in the binary representation)
 - Hamming distance model: $p_i = HD(d_i, d_{i-1}) = HW(d_i \oplus d_{i-1})$

Power Analysis Attack Phases

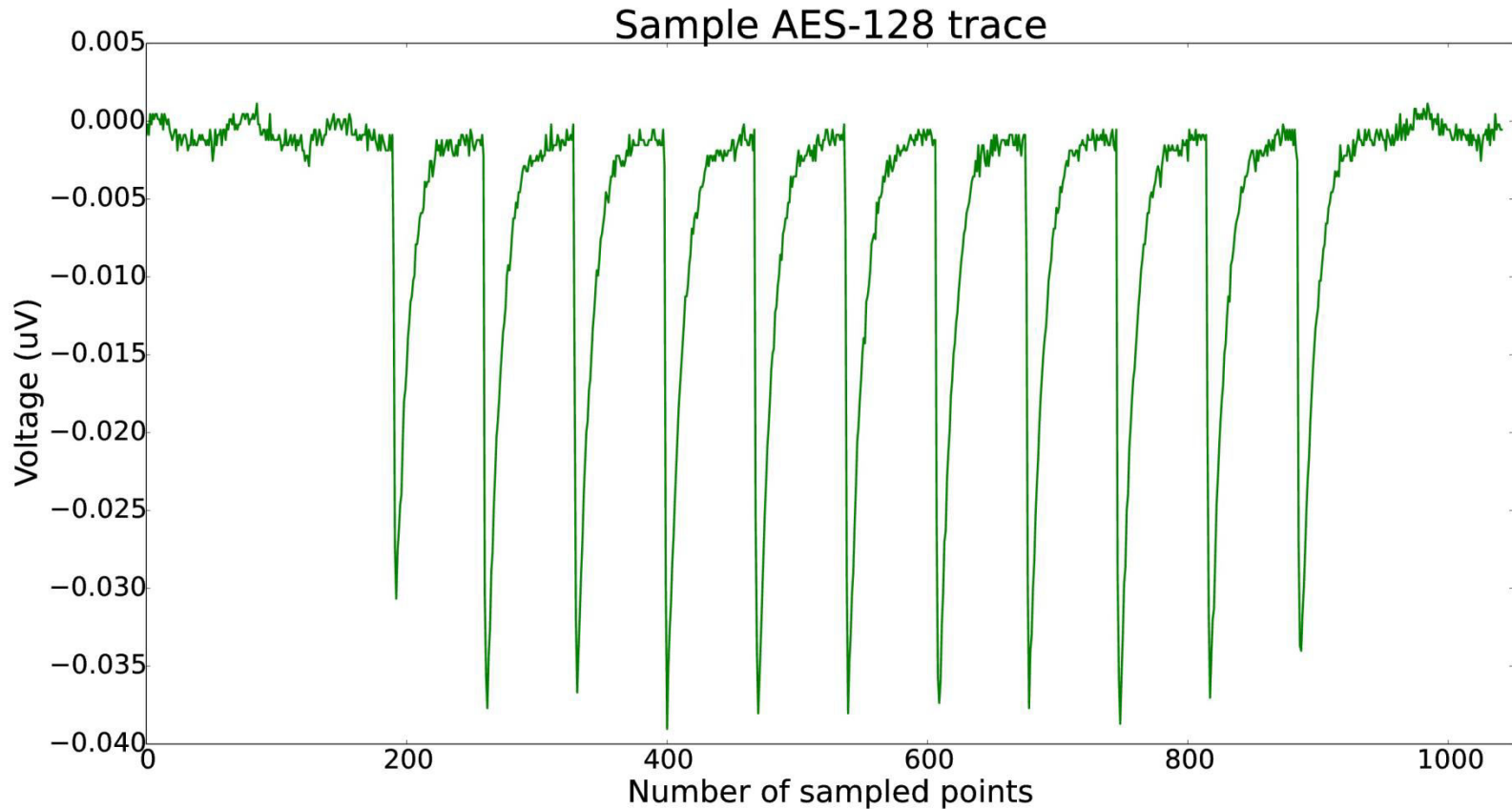
- Measure the circuit's processing time and current consumption to infer what is going on inside it.

■ Side-channel attacks are usually composed of two phases:

- **Interaction (measurement) phase:** interact with the system under attack (SUA) and obtain the physical characteristics of the device
- **Analysis phase:** analyze the gathered information to recover the key



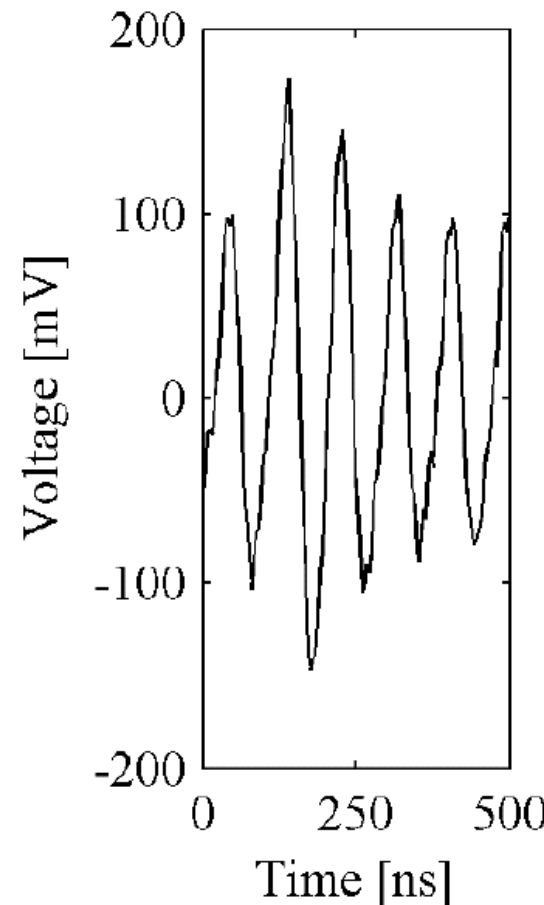
Example of a Power Trace



A power trace consists of T sampling points
(t_1, t_2, \dots, t_T)

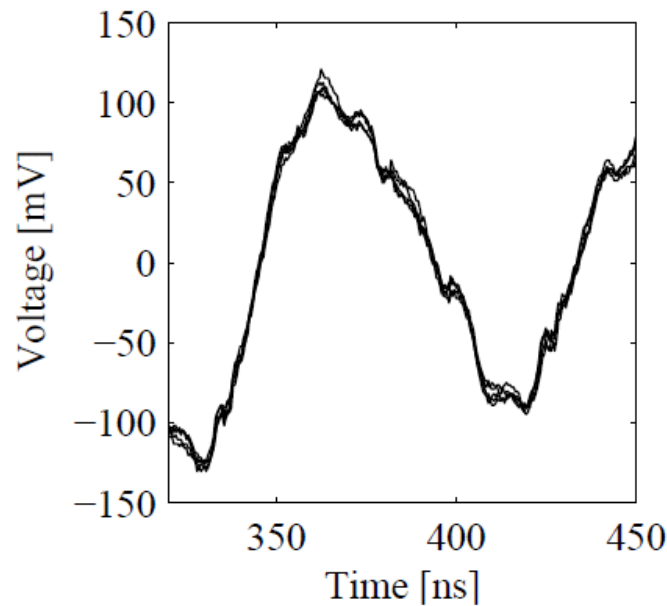
Characteristics of Power Traces

- Individual clock cycles are clearly visible: The clock signal triggers the update of the register content which leads to a lot of switching activity in the circuit
- Height and the shape of the peak depends on the
 - Executed operation (P_{op})
 - The data that is processed (P_{data})



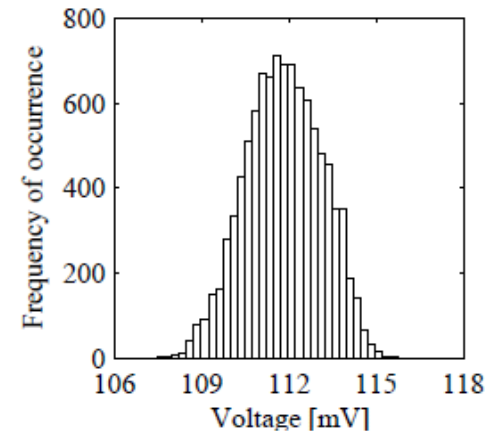
Noise

- Power traces are always associated with noise
 - The measurement of the same device on the same setups leads to different traces
 - The noise that is observed in power traces when performing a fixed operation with fixed data is called electronic noise ($P_{\text{el.noise}}$)



Characteristics of a single point of a trace

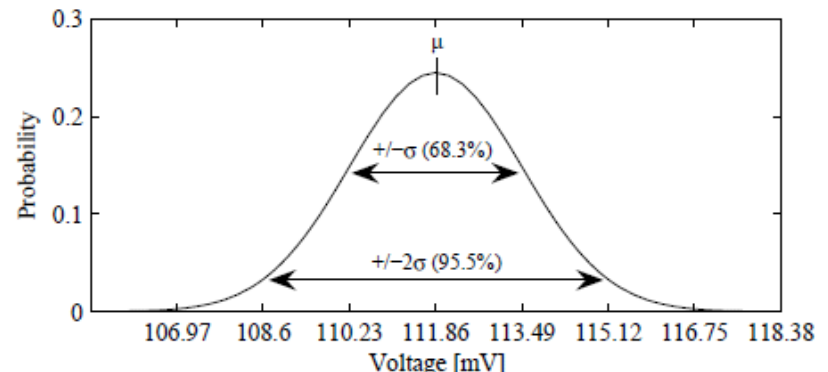
- The distribution of the points (constant data) is typically a normal distribution.
- The normal distribution is fully defined by the mean value μ and the standard deviation σ .



Example:
the distribution of the power consumption at 362 ns:

$$\mu = 111.86$$

$$\sigma = 1.32$$



Model of the power trace

$$P_{\text{total}} = P_{\text{op}} + P_{\text{data}} + P_{\text{el.noise}}$$

- The total power consumption consists of an operation-dependent component, a data-dependent component and noise

Refinement of the model for power analysis attacks:

- In attacks, usually only a small part (P_{exp}) of the power consumption is exploited for attack, we put the power consumptions of the rest of the system into the noise
 - $P_{\text{total}} = P_{\text{exp}} + P_{\text{noise}}$
 - P_{exp} can be HW or HD based, P_{noise} is Gaussian

Pre-processing Side-Channel Information

- A power trace can consist of several millions of sampling points
 - Each sampling point carries information about the performed operation(s) and processed data values
- The information that the attacker is looking for might be present in one single point, in multiple points, in combinations of points, ...
 - The first task that is performed by an attacker is typically to find “interesting” parts of the power trace and to reduce the traces to these parts; this significantly reduces the complexity of all further analysis steps

Principle of Divide-and-conquer Attack

- The divide-and-conquer(D&C) attack attempts at recovering the key by parts, especially for block ciphers
- The idea is that **an observed characteristic can be correlated with a partial key**
 - The partial key should be small enough to enable exhaustive search
- Once a partial key is validated, the process is repeated for finding the remaining keys
- D&C attacks may be iterative or independent

Attack Classification

- Simple vs. differential attacks
 - Simple power attacks directly map the results from a small number of traces of the side-channel to the operation of SUA
 - Differential power attacks exploit the correlation between the data values being processed and the side-channel leakage

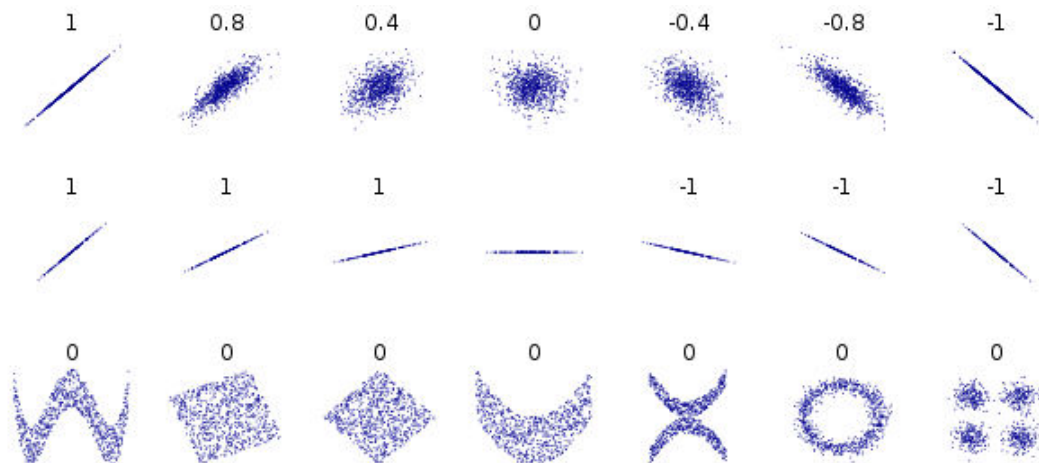
DPA

- The basic idea of the attack is to analyze the power consumption of a device at a fixed moment of time for different plain/ciphertexts of an encryption/decryption
- Power model: $l = \epsilon v + r$, where l is the power leakage, v is the key-dependent select function over an intermediate variable, r is the system noise normally modeled as Gaussian noise:
 $N(c, \sigma)$
 - E.g., for AES, $v = f(S(P \oplus K))$

Correlation Power Analysis

Attack(CPA)

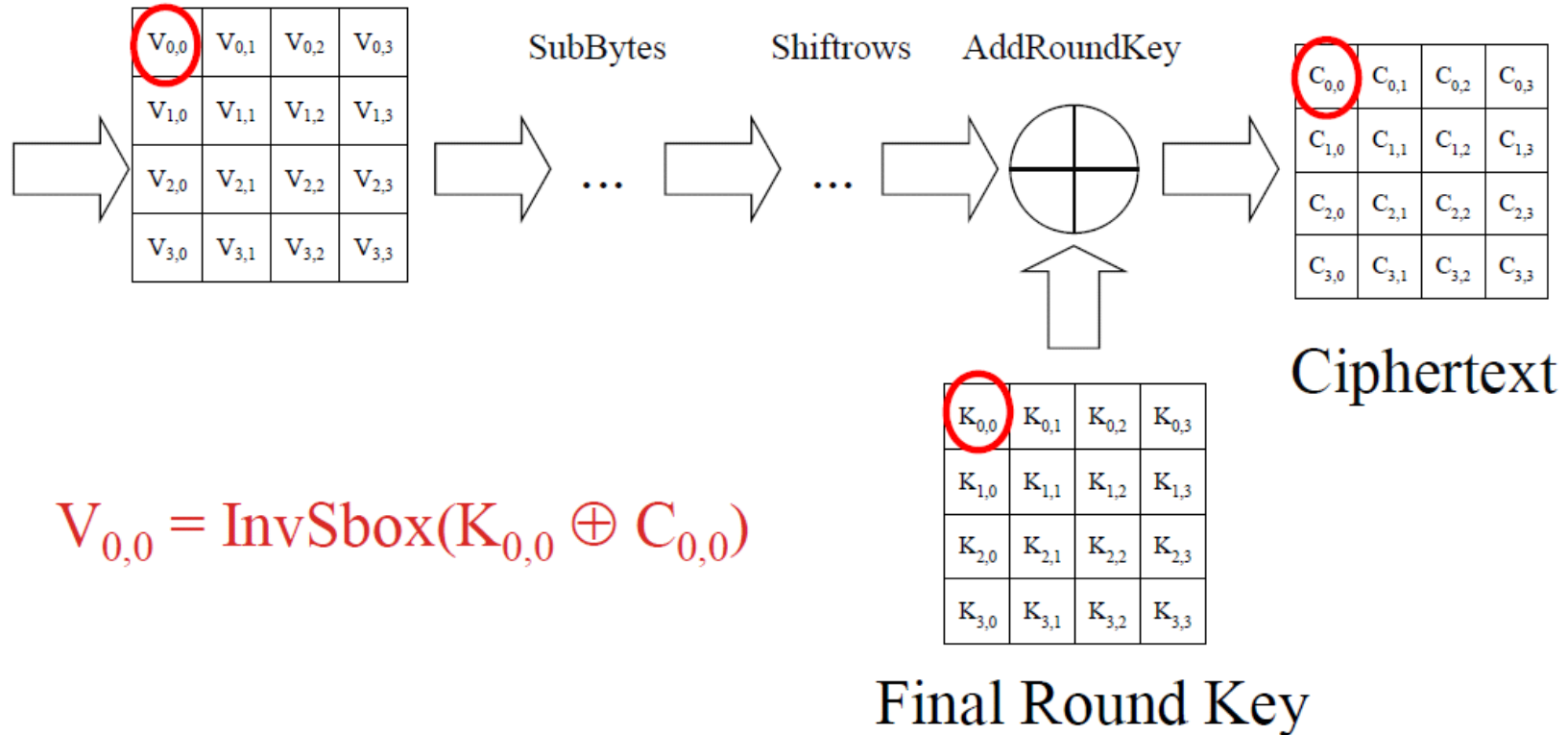
- There exists a strong Pearson correlation between the power leakage and the predicted v value with the correct key: $k_c = \operatorname{argmax} \rho(v|k_g, l)$
- The Pearson correlation coefficient is a well established tool to measure the linear relationship between two random variables:
$$\rho(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y}$$



DoM Based DPA

- When the v is on a single bit, CPA degenerates to DPA, and the statistic to distinguish key guesses is difference-of-means (DoM) instead of Pearson correlation ρ


Exploiting the Data Dependency in the Final Round of AES



DPA on the MSB of an AES Sbox input

in the final round (key guess = 0)

- DoM assumes hamming weight power model

$$\text{MSB}(S^{-1}(C \oplus K))$$


| Sample Nr | Ciphertext | Key Hypothesis | Attacked Value | MSB | Power Consumption |
|-----------|------------|----------------|----------------|-----|-------------------|
| 1 | 0D | 00 | F3 | 1 | H |
| 2 | 95 | 00 | AD | 1 | H |
| 3 | 17 | 00 | 87 | 1 | H |
| 4 | C7 | 00 | 31 | 0 | L |
| 5 | 9B | 00 | E8 | 1 | H |
| 6 | 3B | 00 | 49 | 0 | L |
| 7 | 34 | 00 | 28 | 0 | L |

DPA on the MSB of an AES Sbox input in the final round (key guess =1)

| Sample Nr | Ciphertext | Key Hypothesis | Attacked Value | MSB | Power Consumption |
|--------------|------------|----------------|-------------------|-----|----------------------|
| 1 | 0D | 01 | 81 | 1 | H |
| 2 | 95 | 01 | E7 | 1 | H |
| 3 | 17 | 01 | FF | 1 | H |
| 4 | C7 | 01 | C7 | 1 | H |
| 5 | 9B | 01 | 37 | 0 | L |
| 6 | 3B | 01 | A2 | 1 | H |
| 7 | 34 | 01 | D9 | 1 | H |

DPA on the MSB of an AES Sbox input in the final round (key guess =2)

| Sample Nr | Ciphertext | Key Hypothesis | Attacked Value | MSB | Power Consumption |
|--------------|------------|----------------|-------------------|-----|----------------------|
| 1 | 0D | 02 | FB | 1 | H |
| 2 | 95 | 02 | 85 | 1 | H |
| 3 | 17 | 02 | 2F | 0 | L |
| 4 | C7 | 02 | 07 | 0 | L |
| 5 | 9B | 02 | F9 | 1 | H |
| 6 | 3B | 02 | 5B | 0 | L |
| 7 | 34 | 02 | 24 | 0 | L |

DPA on the MSB of an AES Sbox input in the final round (key guess =3)

| Sample Nr | Ciphertext | Key Hypothesis | Attacked Value | MSB | Power Consumption |
|--------------|------------|----------------|-------------------|-----|----------------------|
| 1 | 0D | 03 | D7 | 1 | H |
| 2 | 95 | 03 | 35 | 0 | L |
| 3 | 17 | 03 | 9B | 1 | H |
| 4 | C7 | 03 | 88 | 1 | H |
| 5 | 9B | 03 | E2 | 1 | H |
| 6 | 3B | 03 | 76 | 0 | L |
| 7 | 34 | 03 | B2 | 1 | H |

DPA on the MSB of an AES Sbox input in the final round (key guess =4)

| Sample Nr | Ciphertext | Key Hypothesis | Attacked Value | MSB | Power Consumption |
|--------------|------------|----------------|-------------------|-----|----------------------|
| 1 | 0D | 04 | 40 | 0 | L |
| 2 | 95 | 04 | AC | 1 | H |
| 3 | 17 | 04 | 82 | 1 | H |
| 4 | C7 | 04 | 33 | 0 | L |
| 5 | 9B | 04 | 6E | 0 | L |
| 6 | 3B | 04 | 25 | 0 | L |
| 7 | 34 | 04 | 08 | 0 | L |

Performing the DPA attack

Time →

↓ Samples

| | | | | |
|----|----|----|----|----|
| 10 | 5 | 4 | 45 | 45 |
| 67 | 56 | 45 | 4 | 8 |
| 37 | 54 | 12 | 45 | 5 |
| 27 | 12 | 69 | 8 | 2 |

Power Consumption

Key Hypotheses →

↓ Samples

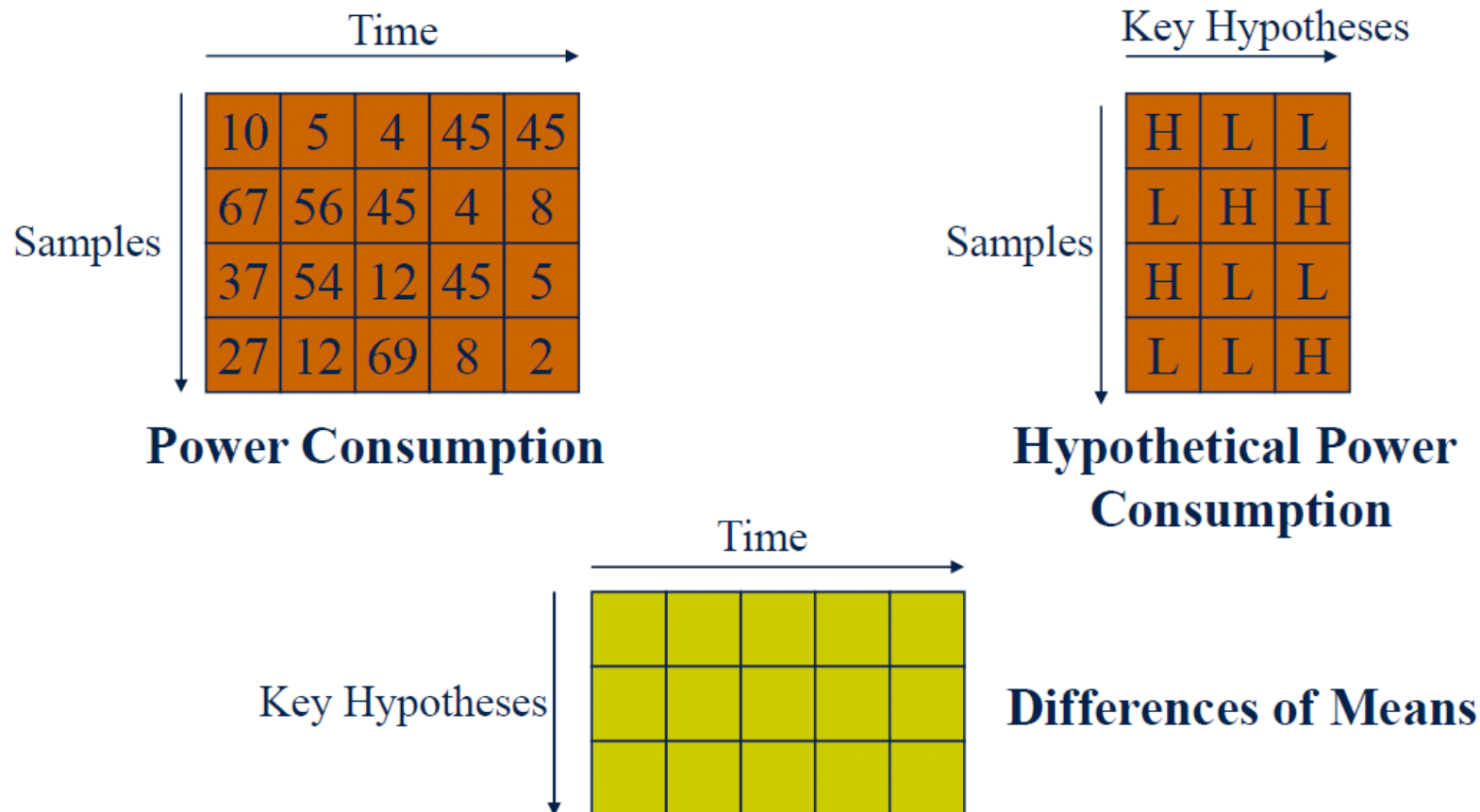
| | | |
|---|---|---|
| H | L | L |
| L | H | H |
| H | L | L |
| L | L | H |

Hypothetical Power Consumption

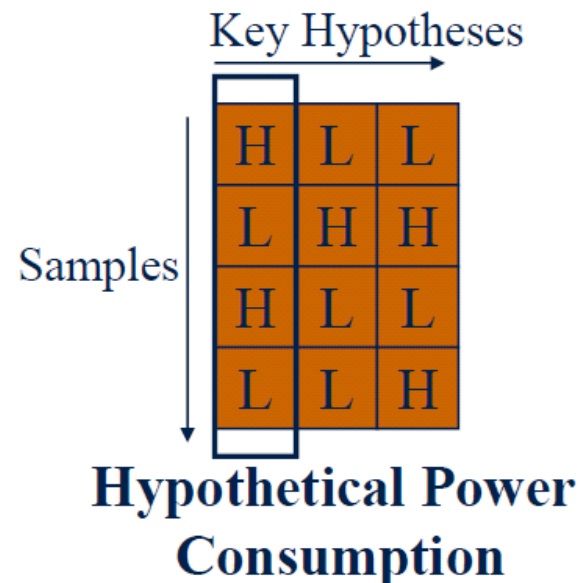
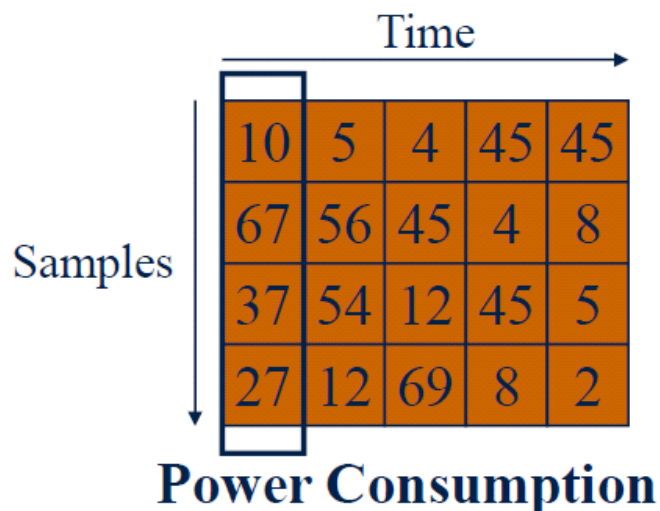
Basic idea of the attack:

1. For every time instance and every key hypothesis calculate the difference between the mean of the samples with high power consumption and the mean of the samples with low power consumption

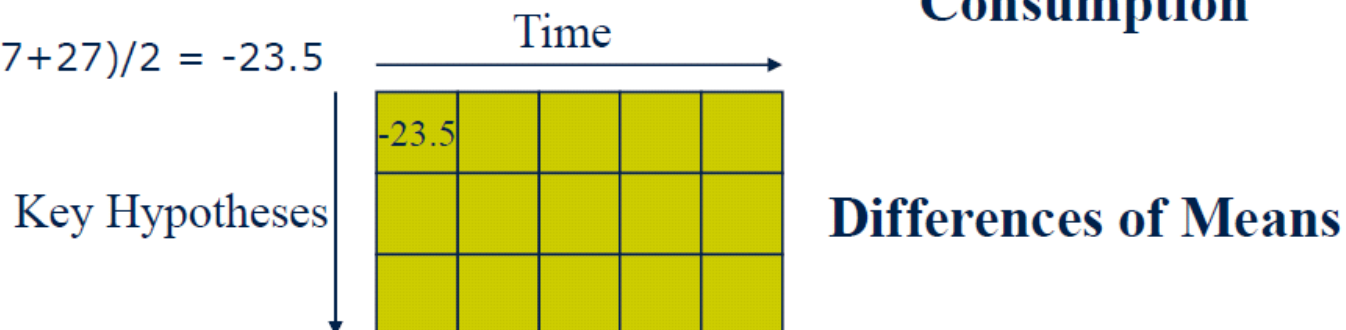
Performing the DPA attack



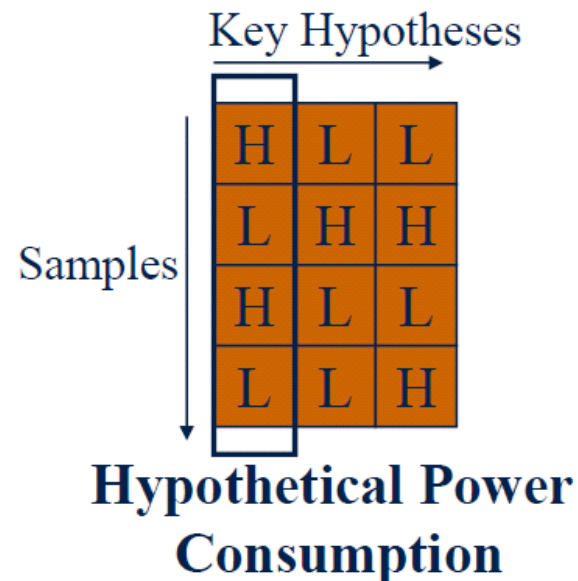
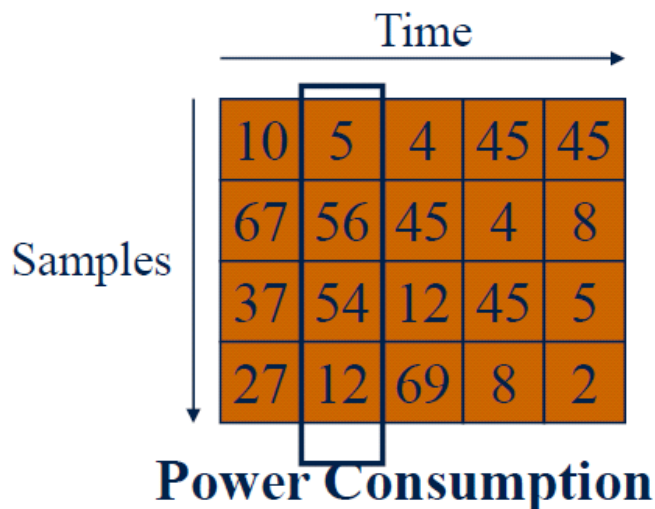
Performing the DPA attack



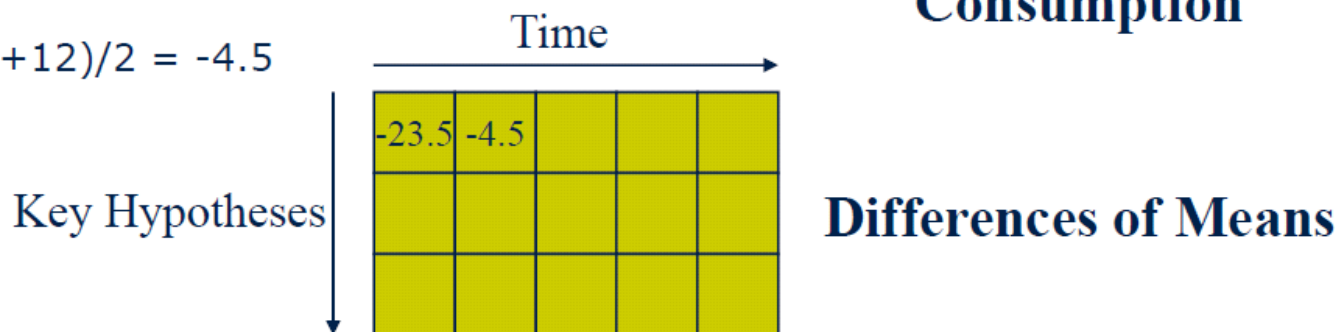
$$(10+37)/2 - (67+27)/2 = -23.5$$



Performing the DPA attack



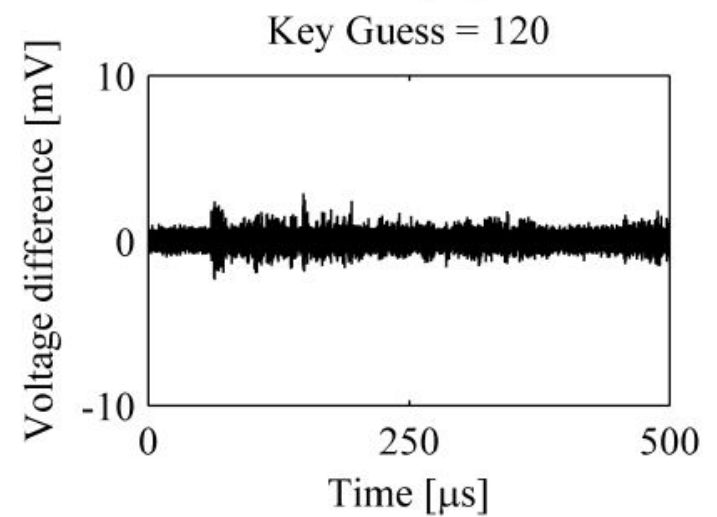
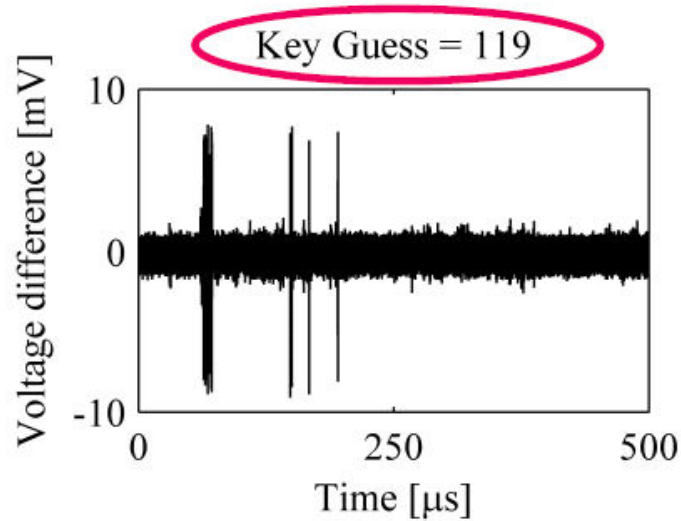
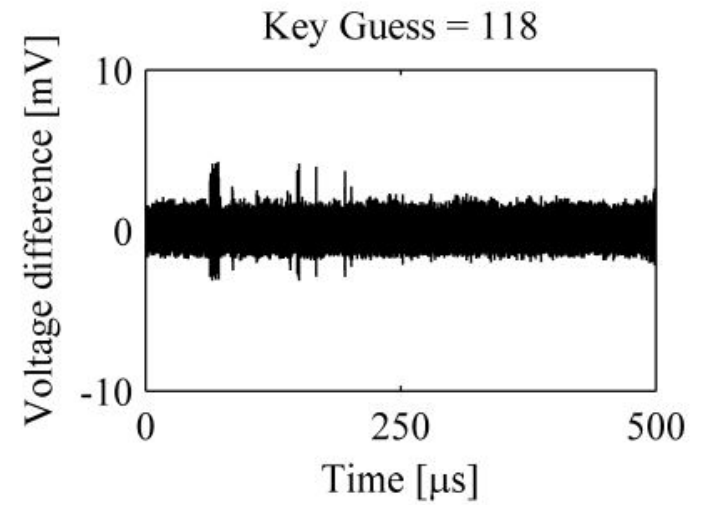
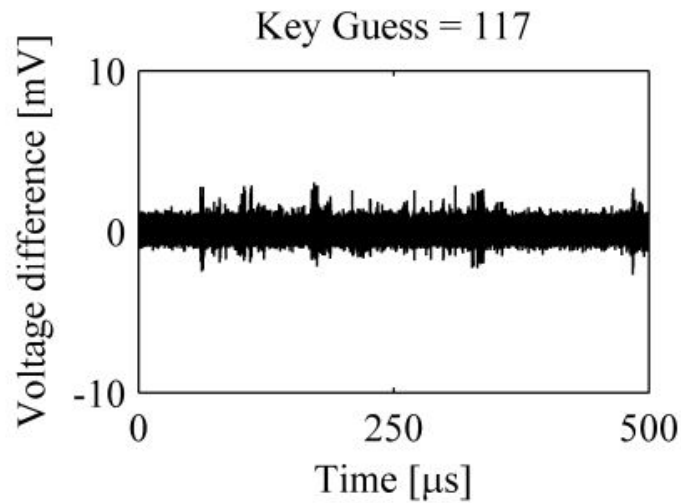
$$(5+54)/2 - (56+12)/2 = -4.5$$



Interpreting the result of a DPA attack

- The result of the attack is a matrix of size
(number of key hypotheses) x (length of the power trace)
- Notation:
 - t_c : column corresponding to the time index when the intermediate value v is processed
 - k_c : row corresponding to the key that is used by the device
- Observations about the elements of the result matrix
 - $t=t_c$ and $k=k_c$: The processing of different values of v leads to different power consumption; Hence a power difference is observable in case $t=t_c$ and $k=k_c$
 - $t \neq t_c$: At all other time instances than t_c operations are performed that are not related to v . Hence, no power difference is observable in case $t \neq t_c$
 - $k \neq k_c$: For all wrong key hypotheses, wrong values v are calculated. These values are largely unrelated to the value that is processed in the device. Hence, only much smaller differences than in case of $(t=t_c \text{ and } k=k_c)$ are observable

Results



Summary

- The location of the maximum in the result matrix reveals t_c and k_c
- Multiple peaks indicate that the attacked value is processed multiple times
- Differential power analysis attacks work, if
 - the number of key hypotheses is small
 - the power consumption of the attacked intermediate result of the algorithm is different for different data values
 - the power traces are aligned

Correlation Power Attack (CPA)

- So far we have performed DPA attacks by calculating DoM
- Drawback of this approach: The attacker is limited to binary power models
 - This is no problem if only a single bit is attacked
 - But when attacking multiple bits the attacker can often formulate more complex power models

Definition of the correlation

- Revisit Pearson correlation:

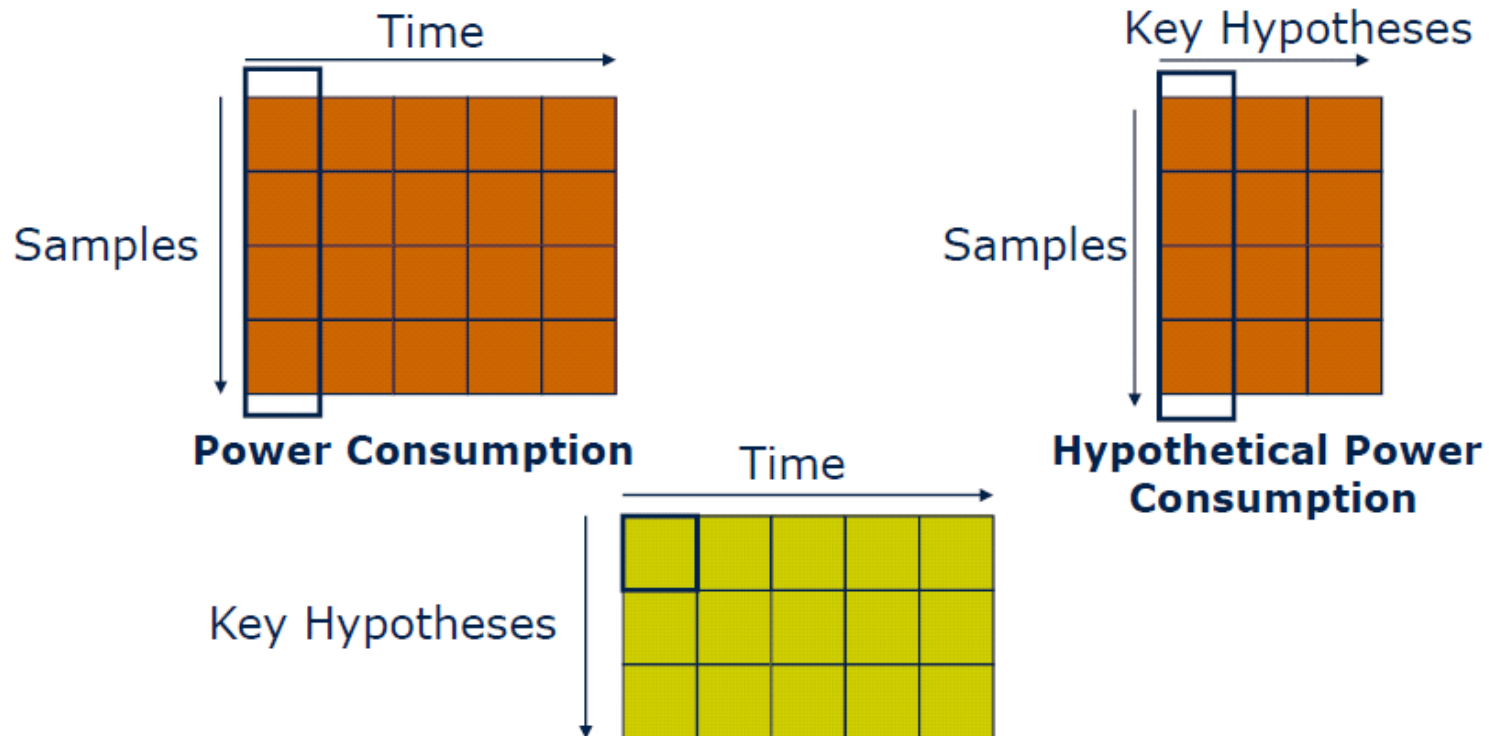
$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}}$$

- It holds that

- $-1 \leq \rho(X, Y) \leq 1$
- if X and Y are independent $\rho(X, Y) = 0$
- if $Y = aX + b$ it holds that $\rho(X, Y) = 1$
- if $Y = -aX + b$ it holds that $\rho(X, Y) = -1$
- $\rho(X, Y)$ is invariant to changes in location and scale $\rho(aX + b, cY + d) = \rho(X, Y)$

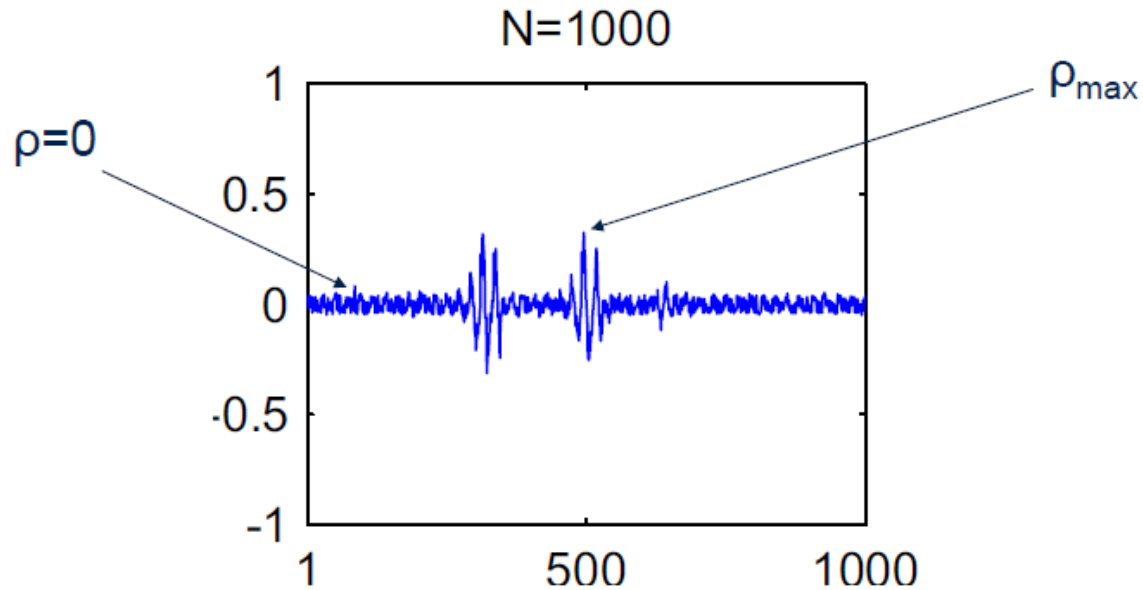
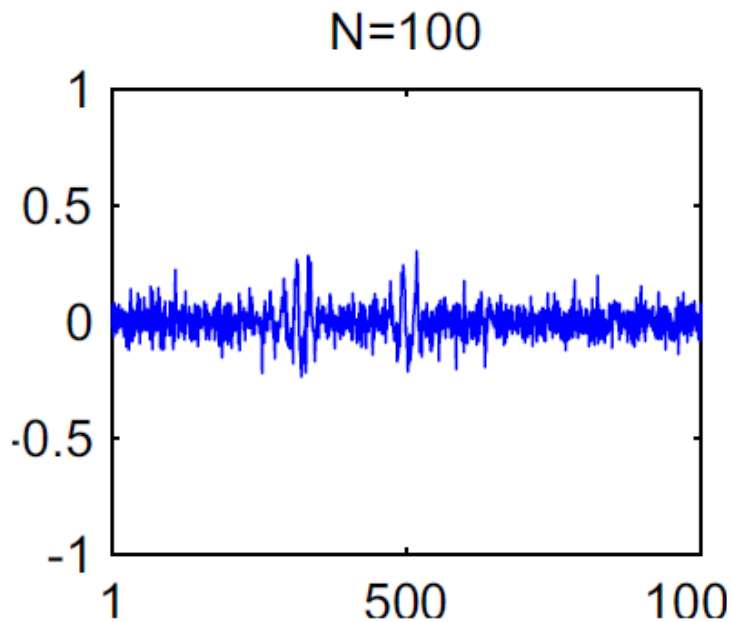
Performing CPA

- The correlation coefficient is used to determine the linear dependency between each column of the trace matrix T and each column of the hypothetical power consumption H (multi-bit); essentially each column is viewed as sample of realizations of a random variable

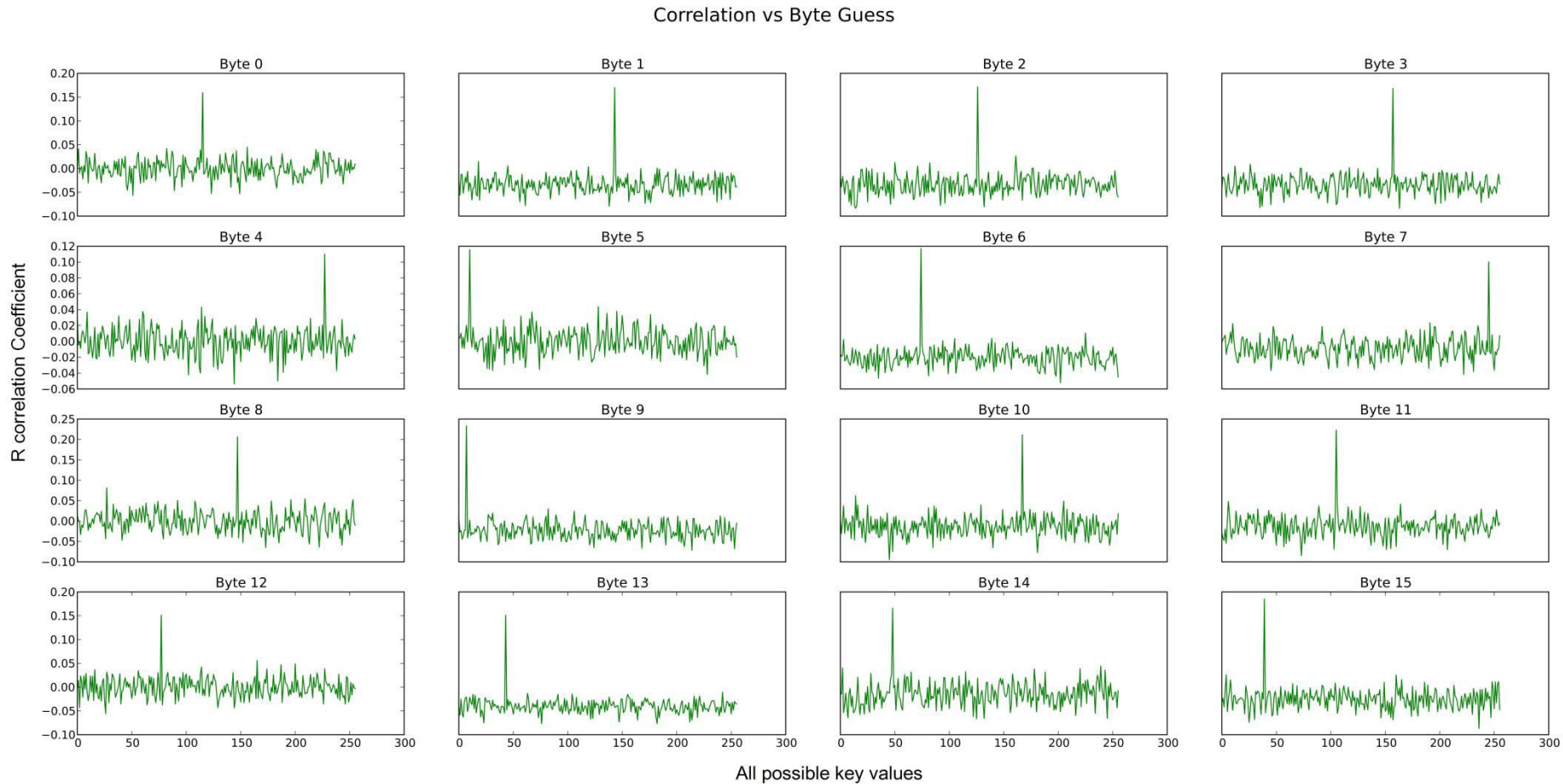


Results for CPA

- The more traces are used, the better is the estimation of the correlation coefficient



FPGA Attack Results

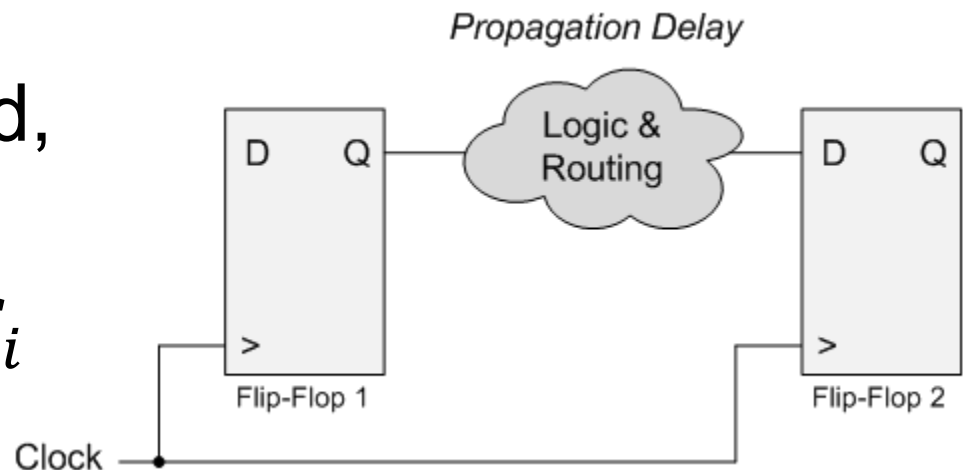


FPGA Implementation of AES

- 128-bit AES takes 11 cycles
- One cycle is for one round
- One round operation: 4 steps in the combinational circuits, input and output registers are the same
- What power model?
- If attack the last round,

$$HD = S^{-1}(C_j \oplus K_j) \oplus C_i$$

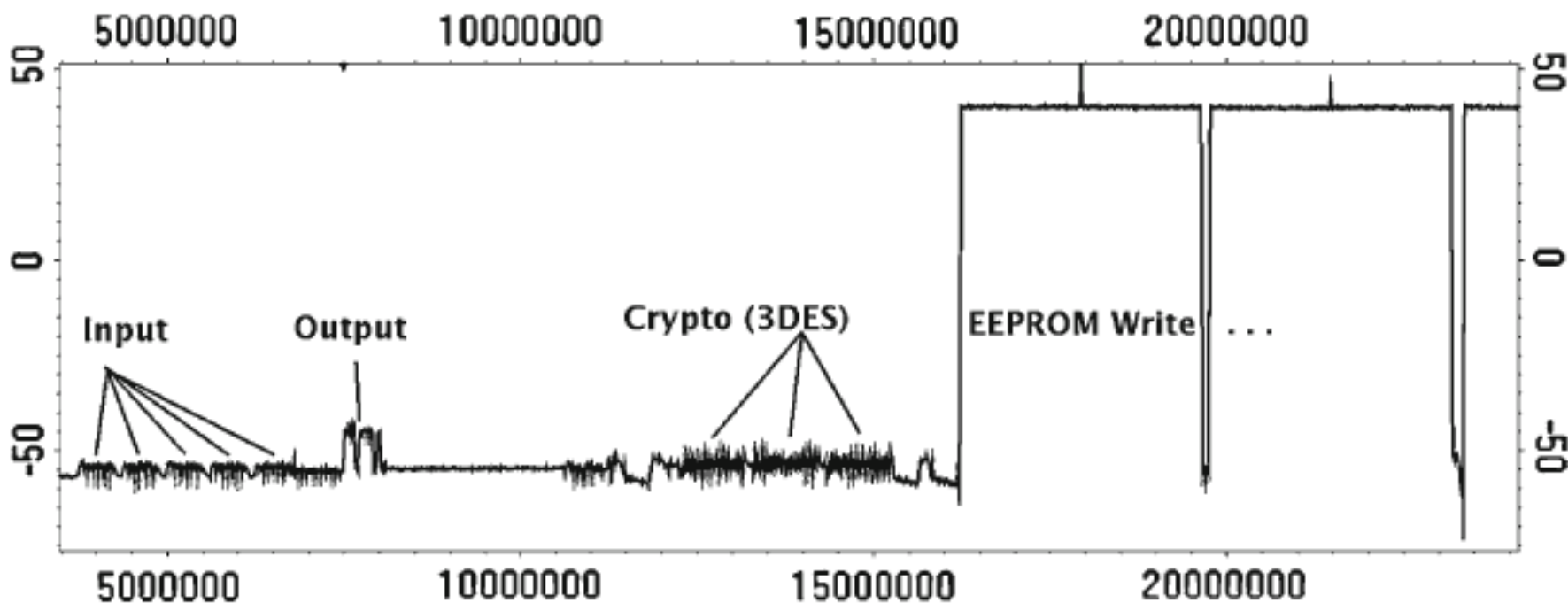
$$HW = S^{-1}(C_j \oplus K_j)$$



Simple Power Analysis (SPA)

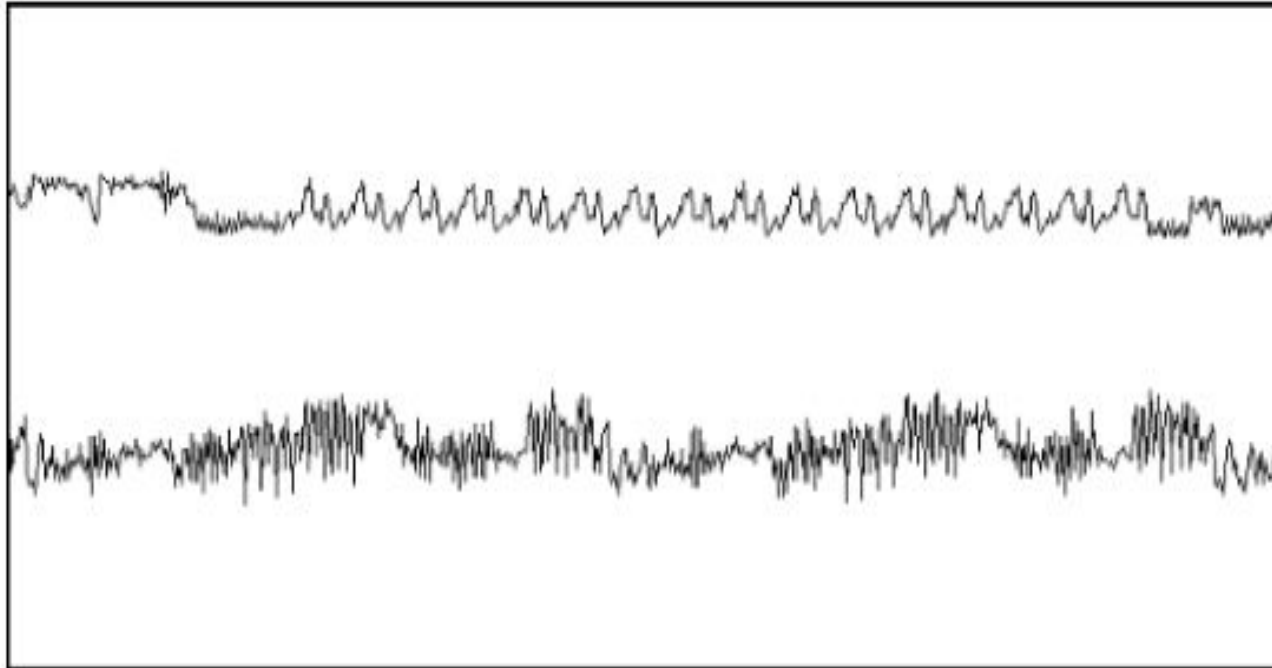
- Originally proposed by Paul Kocher, 1996
- Monitor the device's power consumption to deduce information about data and operation
- Example: SPA on 3DES
- Summary of DES – a block cipher
 - a product cipher
 - 16 rounds iterations
 - substitutions (for confusion)
 - permutations (for diffusion)
 - Each round has a *round key*
 - Generated from the user-supplied key
- SPA targets variable instruction flow
 - SPA is effective for PKE

SPA on 3DES



- From left to right
 - Arrival of input data to the device
 - The output of a single byte
 - A 3-DES operation
 - A series of EEPROM writes
- The power trace can reveal the instruction sequence

DES (cont'd)



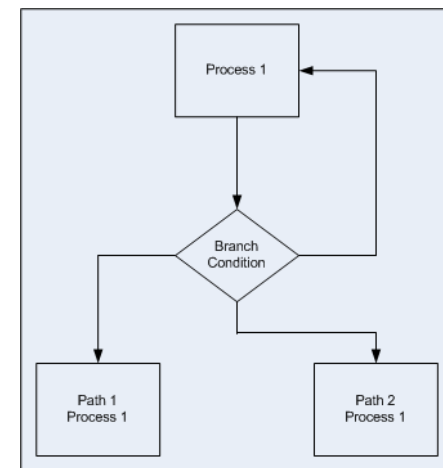
- The upper trace – entire encryption, including the initial phase, 16 DES rounds, and the initial permutation
- The lower trace – detailed view of the second and third rounds

SPA on RSA

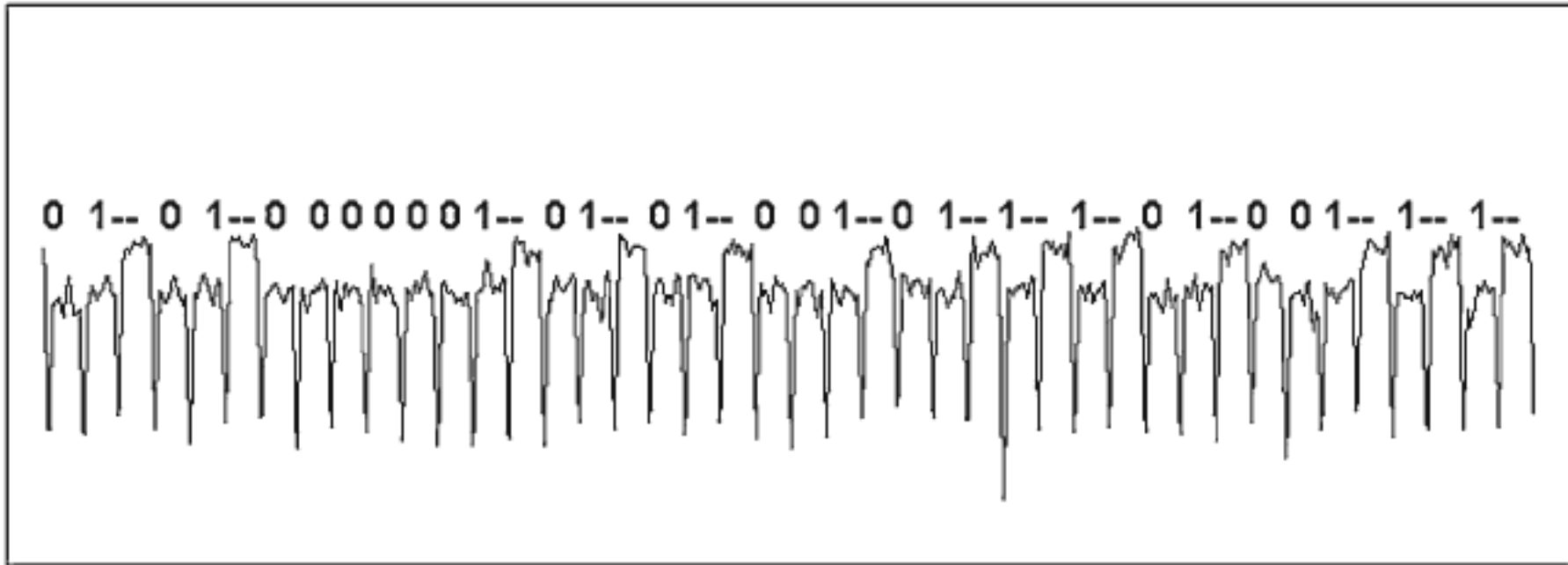
- Example: Modular exponentiation in RSA is often implemented by square and multiply algorithm
 - Typically the square operation is implemented differently compared with the multiply (for speed purposes)
 - Then, the power trace of the exponentiation can directly yields the corresponding value
- All programs involving conditional branching based on the key values are at risk!

```
exp1(M, e, N)
{
  R = M
  for (i = n-2 down to 0)
  {
    R = R2 mod N
    if (ith bit of e is a 1)
      R = R · M mod N
  }
  return R
}
```

square and multiply
algorithm



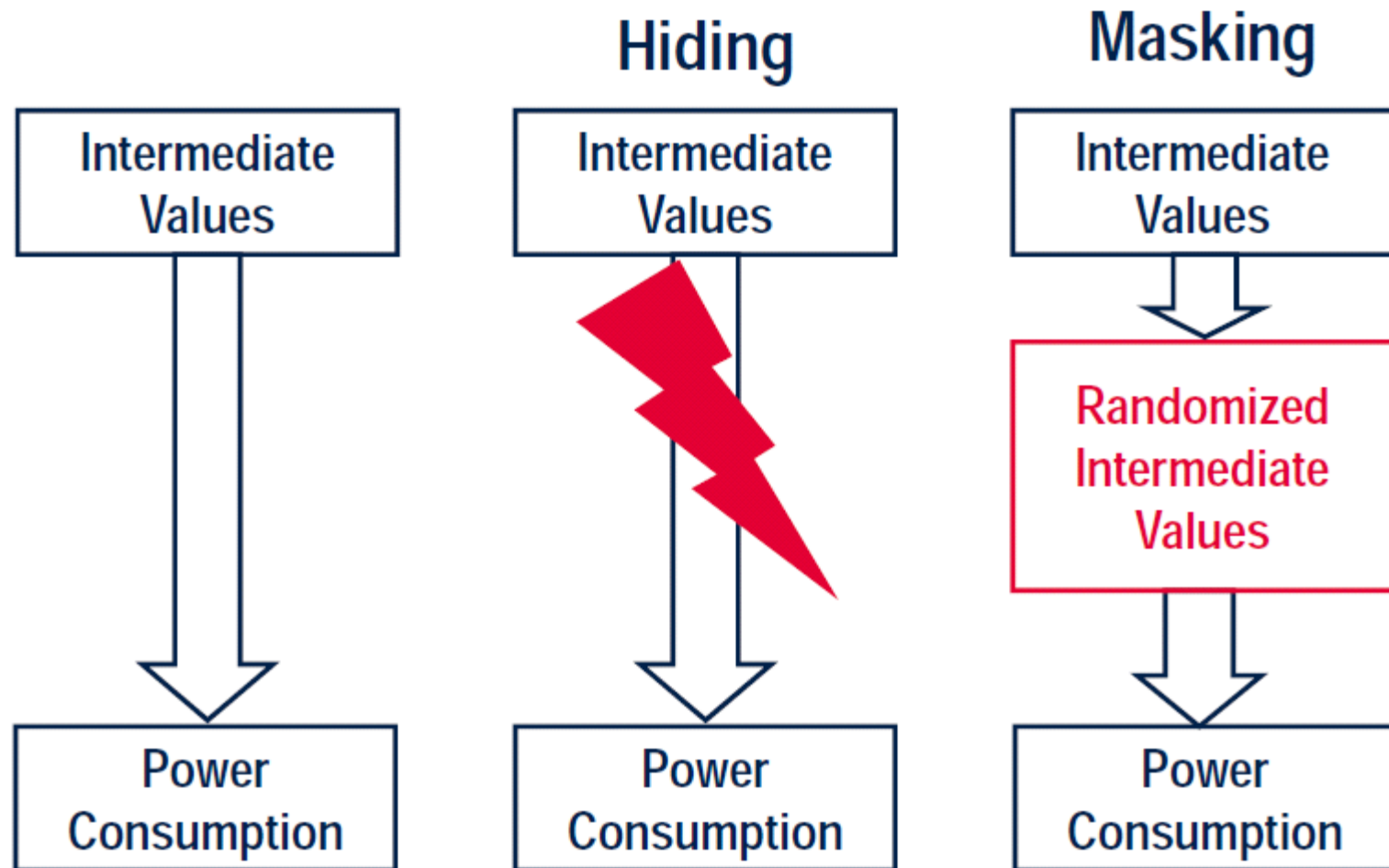
SPA leaks from an RSA Imple.



General Countermeasures

- Making the power consumption independent of the processed intermediate data
 - Hiding -- reduce the SNR of the power channel
 - Increase the noise: Noise Generators
 - Reduce the signal: Balanced Logic Styles, Asynchronous Logic, Low Power Design and Shielding
 - Masking/Blinding -- remove the correlation between the input data and the side-channel emissions from intermediate nodes in the functional block
- Protocol-level countermeasure: switch the key frequently enough such that the attacker does not learn enough information about the key to break the system

Hiding and Masking

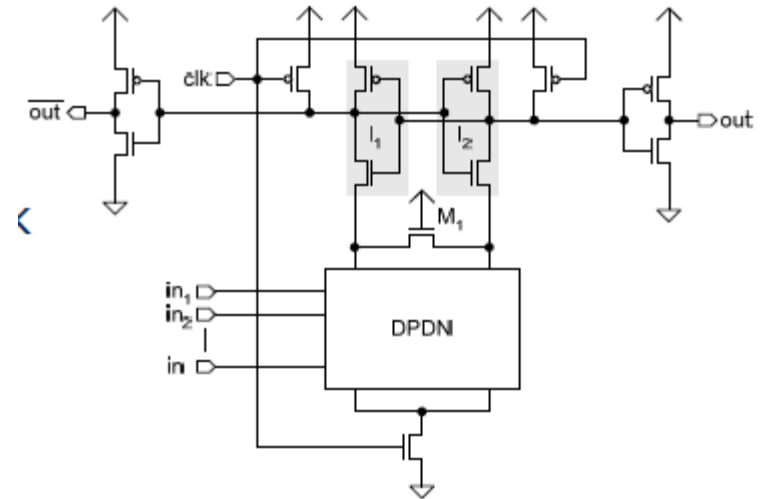


Overview of Countermeasures

| | Hiding | Masking |
|-----------------------|-----------------------------------|--------------------------------|
| Software | Choice of Instructions, Shuffling | Masked software implementation |
| Hardware Architecture | Noise Engines, Shuffling | Masked hardware implementation |
| Cell Level | Dual-rail pre-charge logic styles | Masked logic styles |

Dual-Rail Pre-charge Logic Style

- The system switches between the evaluation phase and the pre-charge phase
 - Pre-charge phase: both wires are set to 1
 - Evaluation phase: one of the wires switches to 0.
- The switching between evaluation and pre-charge phase leads to a constant number of switching events at the output of each gate
- For a constant power consumption it is necessary that the wires are pairwise balanced



Masked logic styles

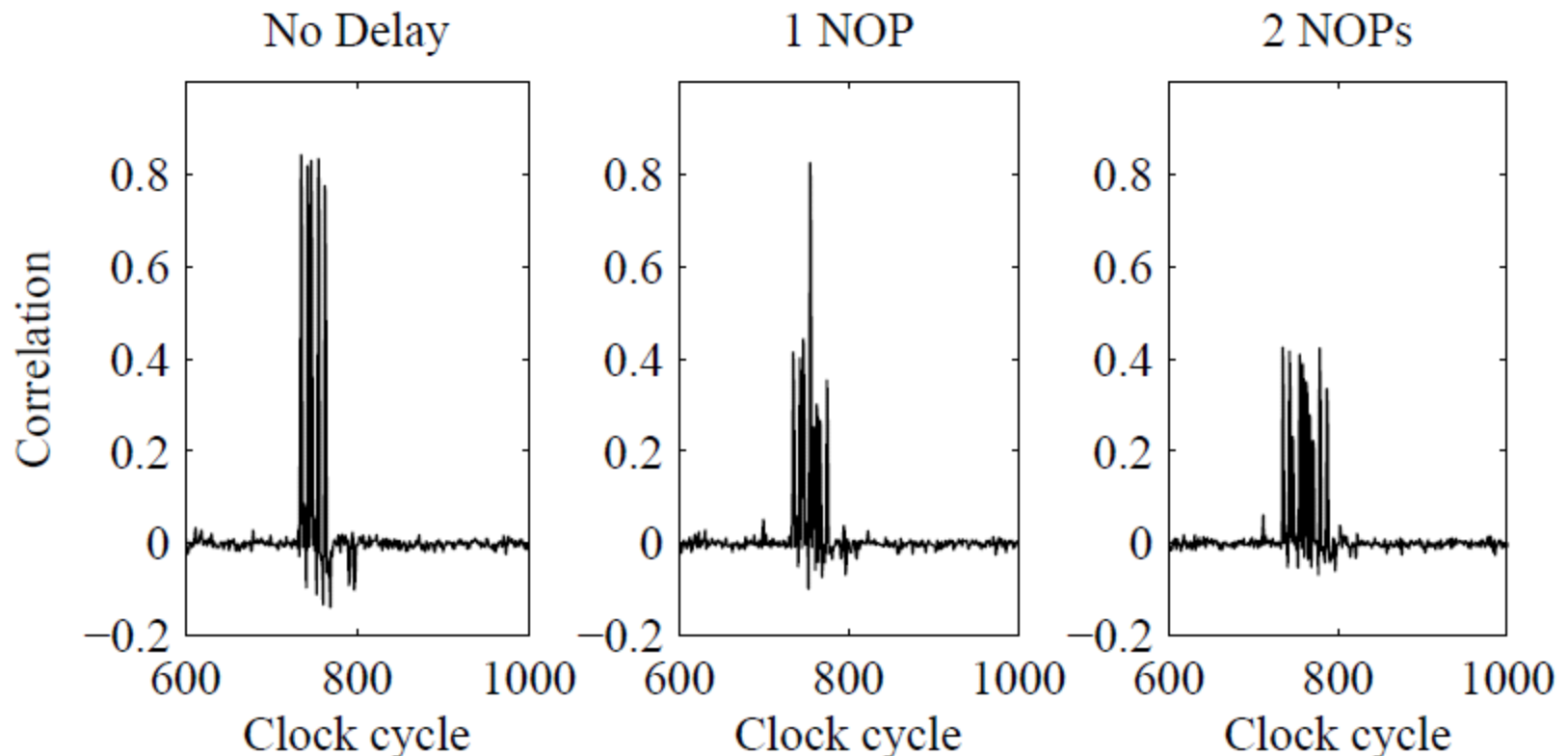
- The goal is to prevent the need to balance the two complementary wires
- Masked logic styles use a random value to randomize all outputs of the gate in a circuit
- More expensive (area and power) than dual-rail pre-charge logic styles

Hiding countermeasures in software

Altering the power consumption of a device is only possible to a limited degree in software

- Choice of instructions: choose instructions whose power consumption is not strongly correlated to the processed data
- Noise: activate additional components on the chip, which consume a significant amount of power and hence cause noise
- Shuffling and random delays: randomly change the sequence of operations of instructions

Example: Attack on the output of an AES SBox



Note: the attacked operation has a certain length on it's own. If not enough delay is inserted, there is no effect on the maximum correlation. In the concrete case, at least two NOP operations need to be inserted.

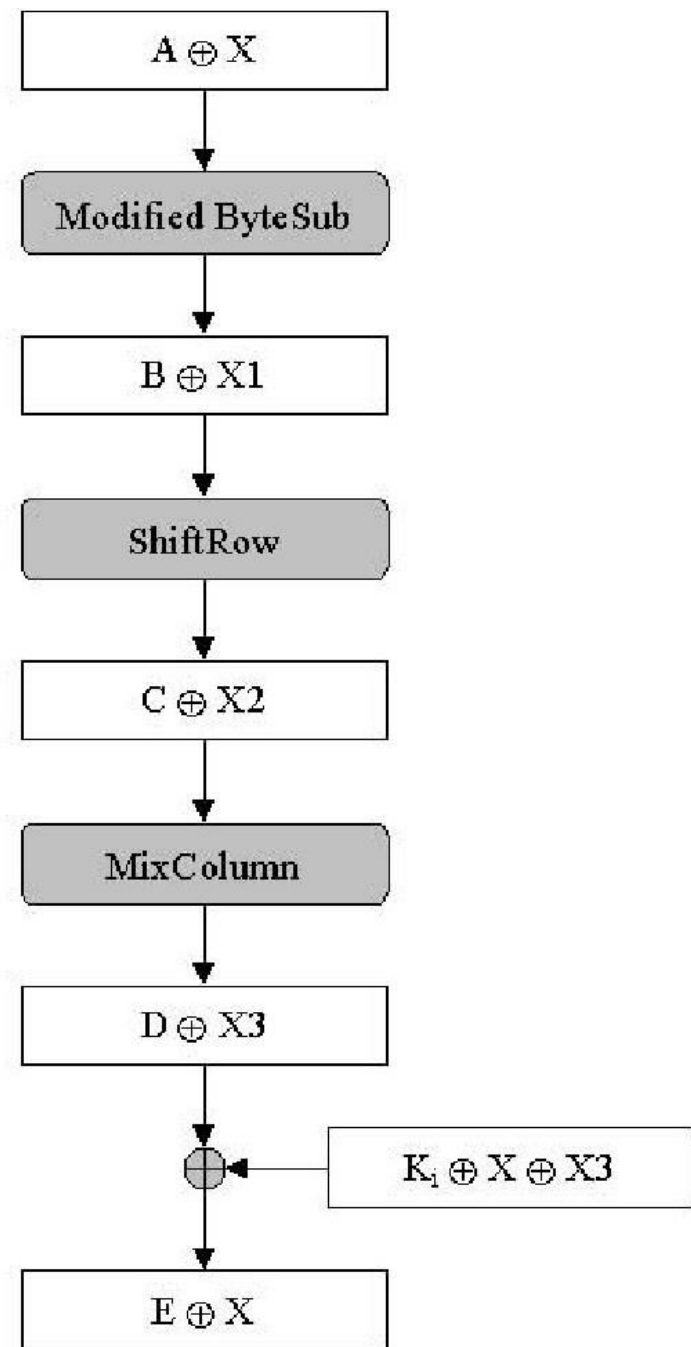
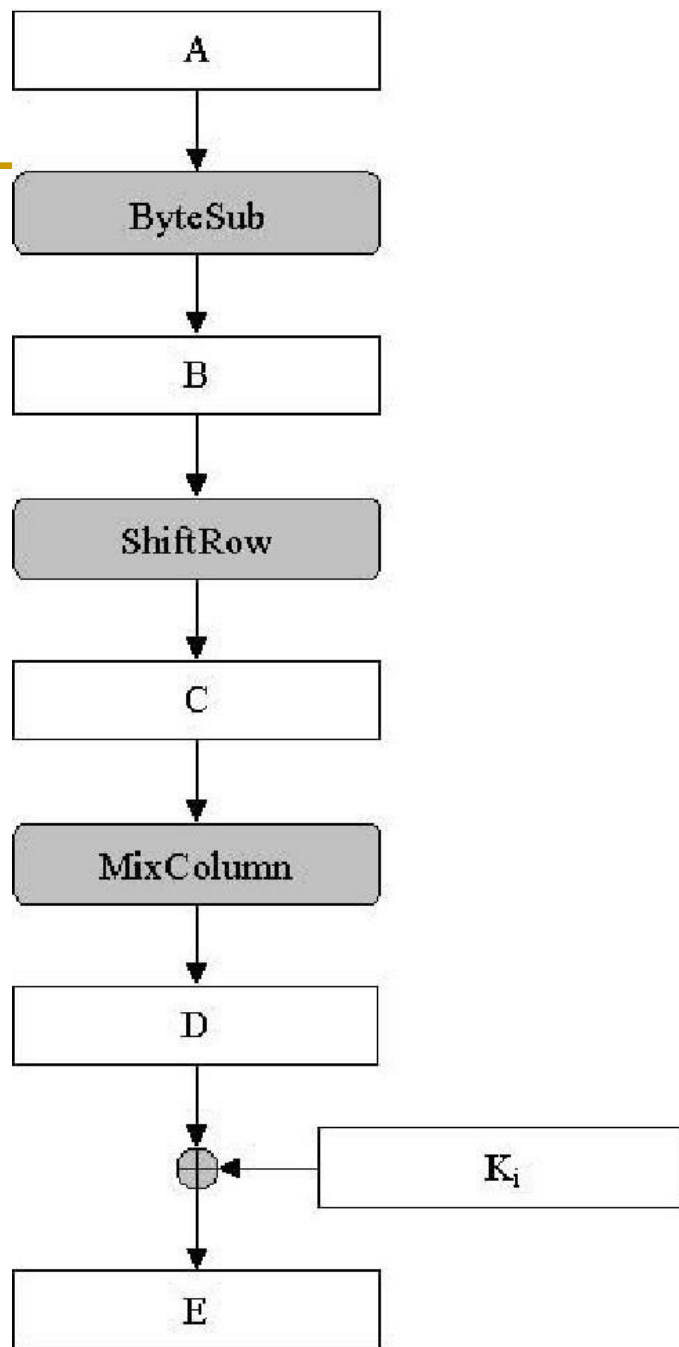
Masking in software

- Can be implemented easily for linear operations
- The most challenging part are the non-linear operations (for both HW masking and SW masking), expensive (RAM and performance)
- Masking requires a random number generator

Masking is the most commonly used and most effective SW countermeasures against power analysis attacks in practice

Masking different operations in AES

- Masking of linear function $f(x)$ is easy; it holds that
 - $f(d) = f(d_m) \text{ XOR } f(m_d)$
 - Hence, the function simply needs to be applied on the masked data and the mask
- Masking of non-linear operations like Sbox is difficult
 - $O=S(i) = S_m(i \text{ XOR } m_i) \text{ XOR } m_o$
 - Mask needs to be updated frequently – ideally for each encryption



References

- [1] Paul C. Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi: *Introduction to differential power analysis*. Journal of Cryptographic Engineering, Volume 1, Springer Verlag, 2011
- [2] P. Kocher, J. Jaffe, and B. Jun (1999): *Differential power analysis*. In: 19th Annual International Cryptology Conference (CRYPTO), vol 2139. Springer-Verlag, Berlin, Heidelberg, New York, August 1999
- [3] K. Tiri and I. Verbauwhede (2003) Securing encryption algorithms against DPA at the logic level: next generation smart card. In: CHES 2003, vol LNCS 2779, pp. 125–136

Lab 3 - CPA

- You are given a set of power traces of FPGA AES-128 implementations under different input blocks
- You are told which time point is the leaky one (corresponding to the target data)
- Implement correlation power analysis attack to recover the 128-bit key, be careful in your choice of power model: Hamming weight or Hamming distance model