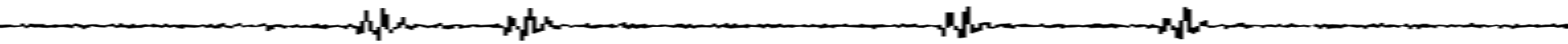
Applied Security

Power Attacks on RSA (4)

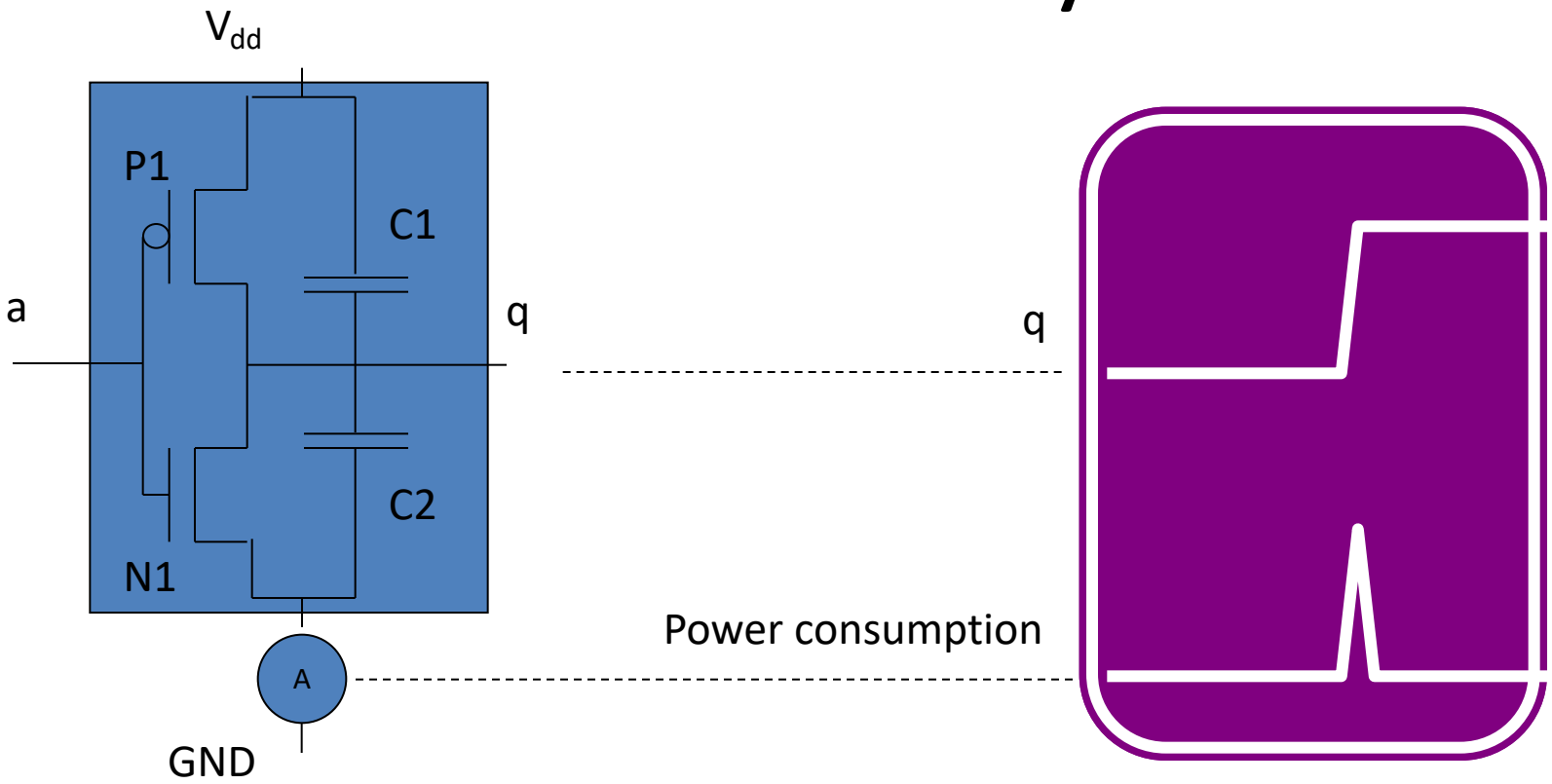
General Overview

- Focus on power (/EM) attacks on RSA
- What is a power attack?
- What might cause power leaks?
- Simple exploitation of leaks
- Differential power attack

Beware: in order to follow the lectures you NEED to be familiar with various cryptographic algorithms and implementation techniques!



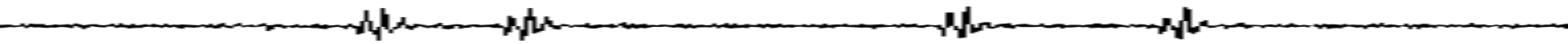
Power analysis



- The power consumption of a CMOS gate depends on the data:
  - q: 0->0 virtually no power cons.
  - q: 1->1 virtually no power cons.
  - q: 0->1 high power cons. (proportional to C2)
  - q: 1->0 high power cons. (proportional to C1)
- Consequently: the power (or EM) profile depends on the data that is processed and the operations used!

Sources for power side channels

- Different operations have different power profiles:
  - Gate level: AND gates doesn‘t need to wait for other inputs if one is zero so needs less power to change
- Architectural differences:
  - Data busses connect different parts of a chip (and memory), they are ‚long‘ and drive different parts of the chip so consume a lot of power
  - Registers hold values, switching causes power side channel
  - Large number multipliers are ‚big‘



Power Analysis Attacks

- A power analysis attack uses power (or EM) leakages to determine information about the secret key
- In case of RSA this would be
  - The secret key used for decryption
  - The secret key used for signing
- A simple power attack typically uses a single observation whereas differential power attacks uses many observations

RSA

- Focus is on decryption:
  - d decryption key, N modulus, c ciphertext, then m = c^d mod N
- To implement the modular exponentiation there exist a variety of implementation options
  - Binary exponentiation, window methods
  - Montgomery multiplication
  - Barrett reduction
  - Etc.

$d=\{d_w,d_{w-1},d_{w-2},\ldots,d_1,d_0\}_2$

m = 1;

For i = w-1 to 0

  m = m • m mod n

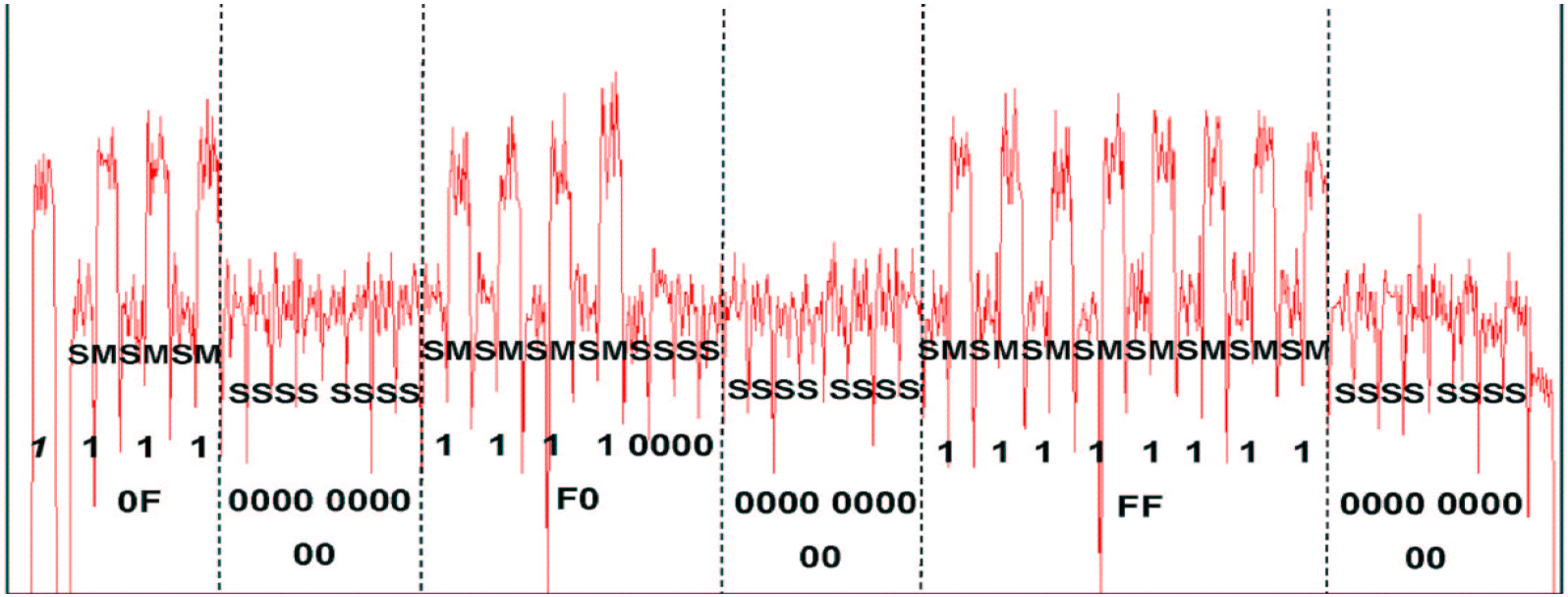
  if (d_i) = 1

    then m = m • c mod N

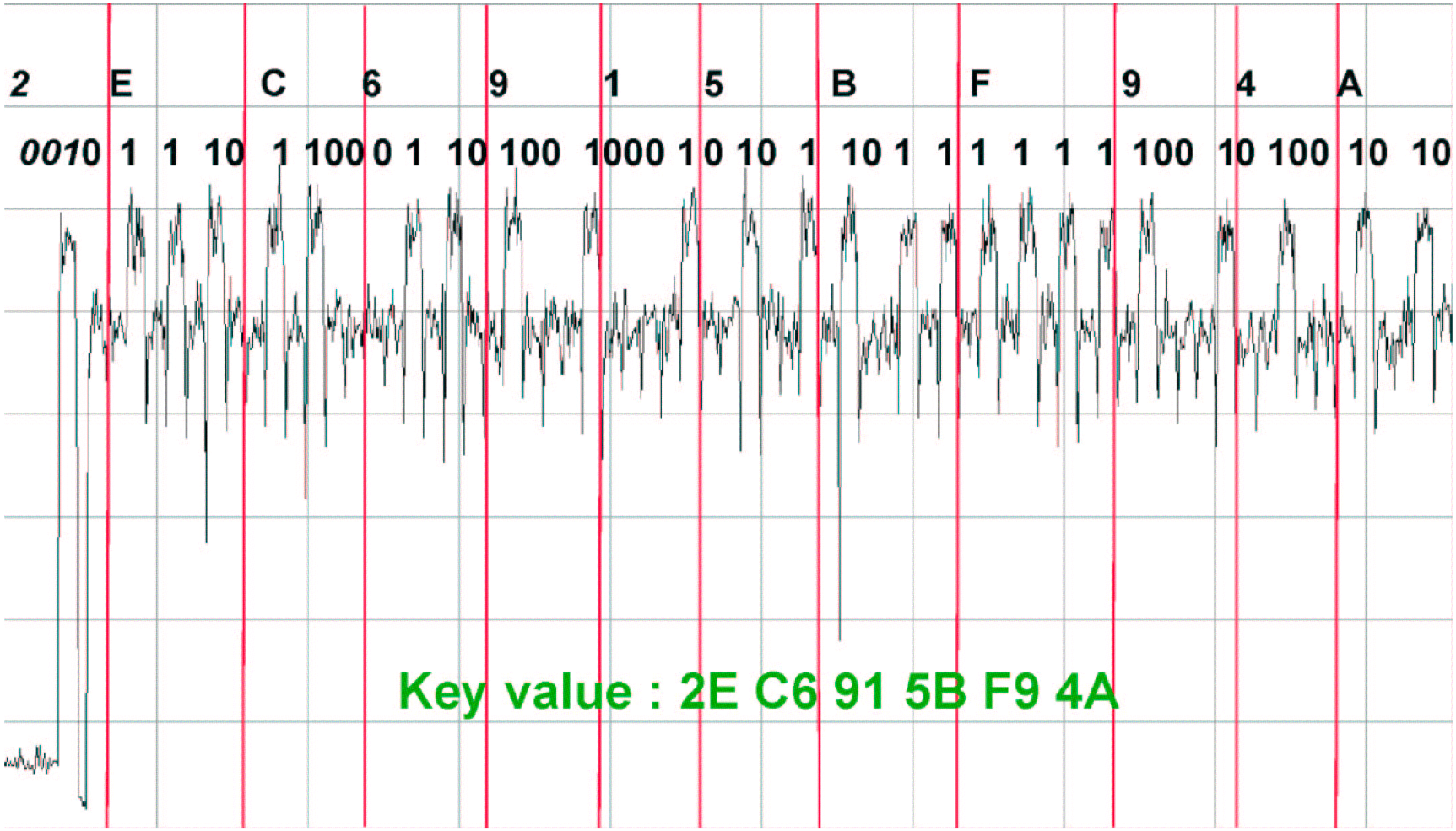
  (endif)

(endfor)

Some power traces with some simple analysis

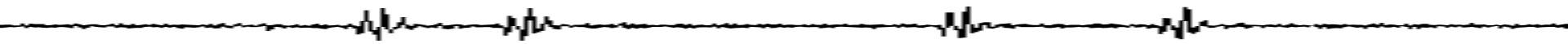


The difference between square and multiply operations is clearly visible in the trace. Consequently, the key value can be directly determined.


2 E C 6 9 1 5 B F 9 4 A
0010 1 1 10 1 100 0 1 10 100 1000 1 0 10 1 10 1 1 1 1 1 1 100 10 100 10 10
Key value : 2E C6 91 5B F9 4A

Visual inspections may also break implementations using windowing alg.

- RSA
  - Clearly, if square operations look different than multiply operations, one can recover the secret key directly!
- Windowing algorithms
  - Scan scalar/exponent not bit-wise but by windows of bits
    - If addition/multiplication by different values looks different than doubling/squaring (whatever other operation), same trick works again!
- However, also small features can lead to SPA attacks!



Slightly more complicated power analysis attacks for RSA

- We can also exploit the encryption operation. So key idea is to use the fact that we can encrypt
  - The same or different data with
  - Known and unknown keys
- Two scenarios described by Messerges
  - No control over PK and SK (i.e. they are fixed), but control over data
  - Control over PK, no control over SK (i.e. it is fixed), limited control over data (i.e. data unknown but fixed)

RSA: Comparing power traces produced with known/unknown exponent (1/2)

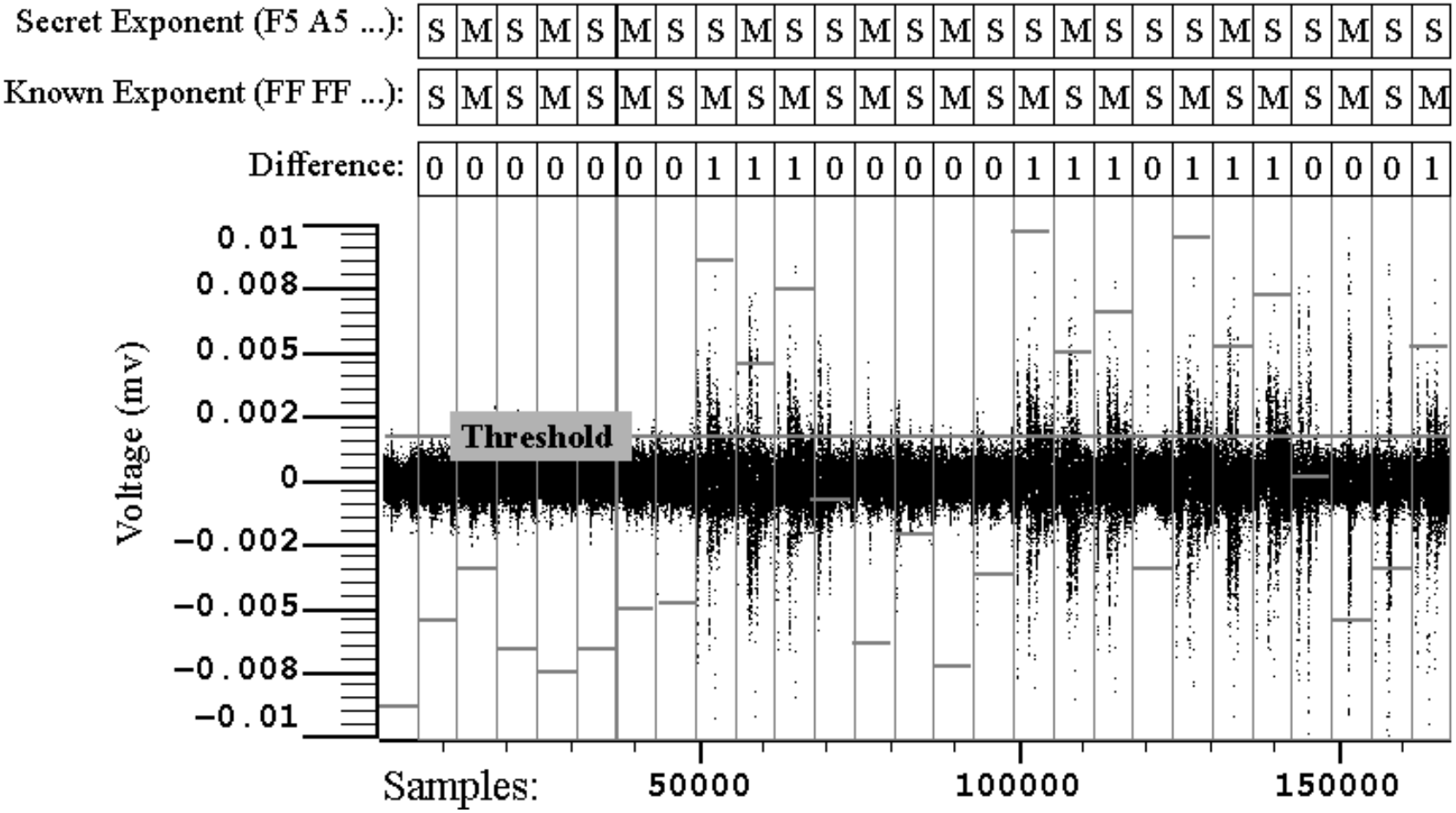
- Suppose the same set of messages are encrypted using a know (chosen by the attacker) and an unknown exponent
  - $C_1^{pk} = m_1 \text{^} e \bmod n$
    $C_1^{sk} = m_1 \text{^} d \bmod n$

- There should be differences in the power traces of C_1^{pk} and C_1^{sk} due to the differences in the exponents
  - These differences are likely to be small so several traces are needed to reveal them

- Compute difference trace d
  - Data dependency averages out
  - Exponent dependency should become visible

$$d = \sum_i t_i^{pk} - \sum_i t_i^{sk}$$

RSA: Comparing power traces produced with known/unknown exponent (2/2)

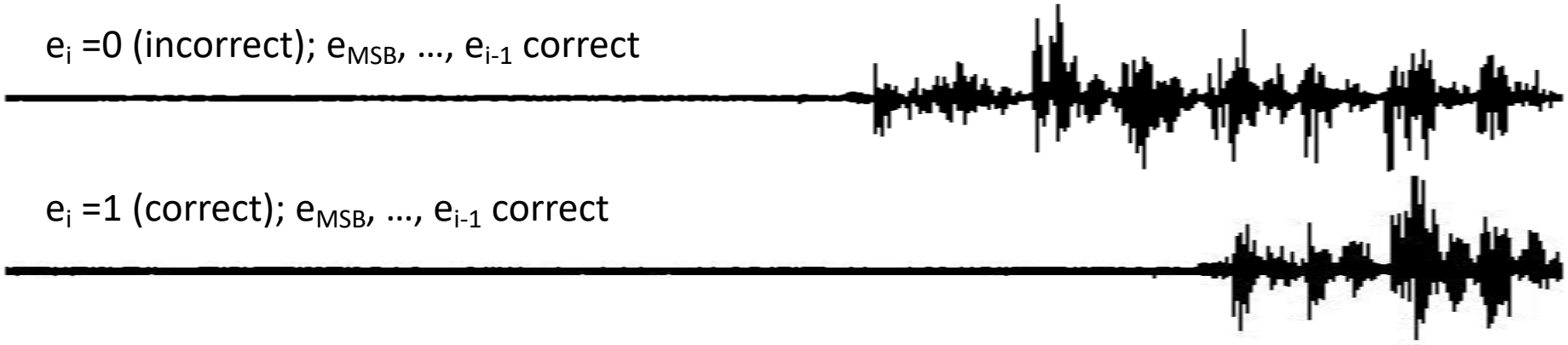


RSA: Comparing power traces produced with known exponents, fixed data (1/2)

- Suppose the attacker may use different public exponents with a single message
  - C^{sk} = m^d mod n
  - C_e^{pk} = m^e mod n

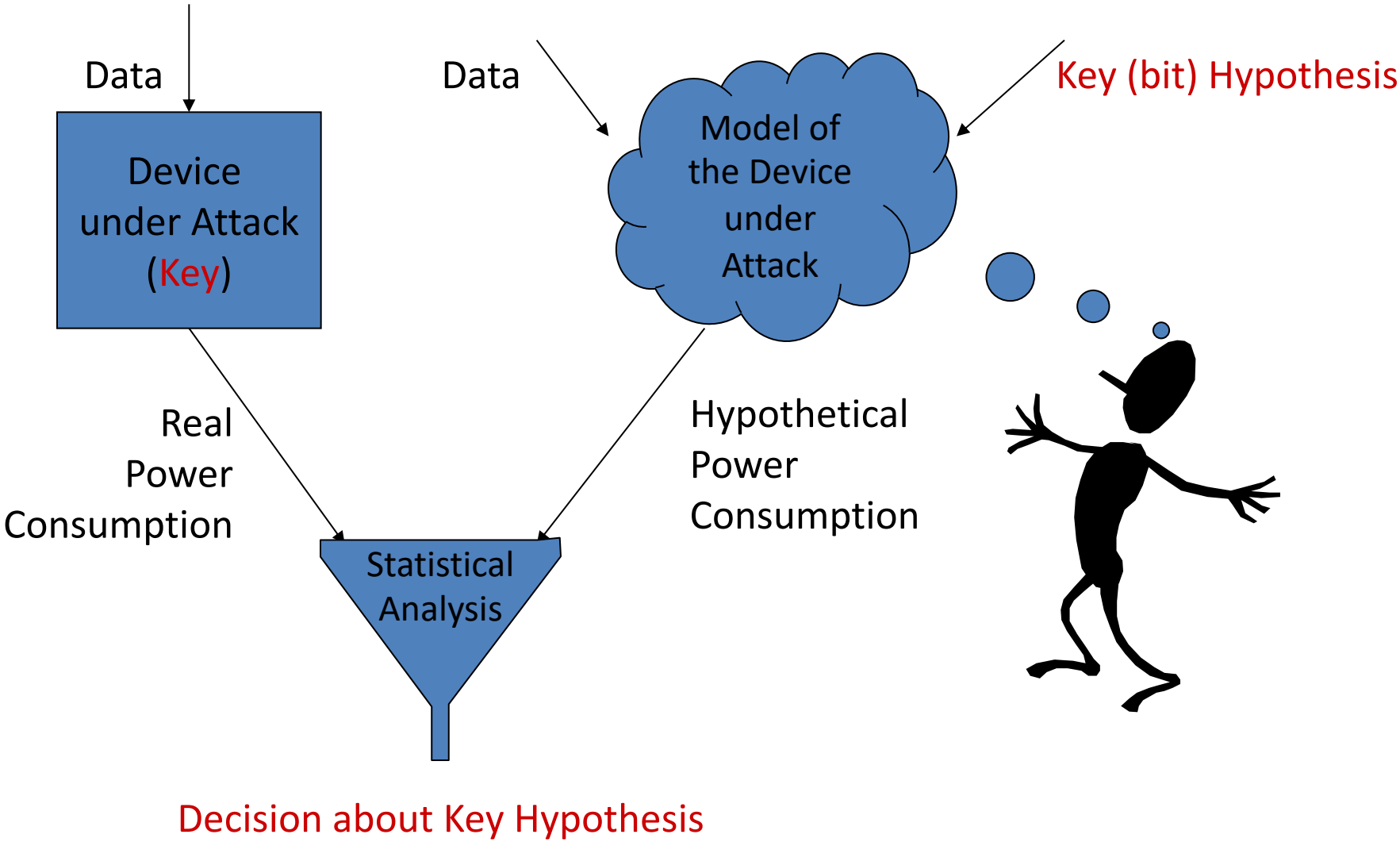
- In the binary algorithm, the same sequence of exponent bits lead to the same intermediate values
  - If first i bits of unknown exponent d and known exponent e coincide, the power traces will be very similar
    - Use averaging to get rid of noise
    - Compute difference trace for e_i=0, and another for e_i=1, the difference trace which stays close to 0 longer indicates correct guess

RSA: Comparing power traces produced with known exponents, fixed data (2/2)



- If more bits are set correctly in the chosen exponent e, the difference trace stays longer close to 0
- Exponent is revealed bit by bit
- The last two examples refer to a technique sometimes called as ‘trace pair analysis’

Principle of a differential power attack



DPA attacks require some knowledge about the implementation

- There are many implementation options for PK algorithms
  - Example: RSA
    - Montgomery multiplication
    - Barrett reduction
    - Binary vs. windowing methods
      - For exponentiation
      - For multiplication
    - …
- In order to predict intermediate values, attacker must know (or be able to narrow down) the implementation details

Case study for RSA: DPA using HW of result of multiplication (squaring)

- Assume we can
  - invoke RSA operations using the unknown private key, and known plaintexts
    - RSA(d,m)→m^d mod n
  - invoke RSA operations using a known public key, and known plaintexts
    - RSA(k,m) →m^k mod n

- If first l bits of k equal first l bits of d then RSA(k,m) is the same as intermediate value at i=n-l in RSA(d,m)

Case study for RSA: DPA using HW of result of multiplication (squaring)

- Select intermediate result, generate inputs
  - Intermediate value of squaring
- Acquire power traces
  - Large number N of power traces, different plaintexts, same key
- Calculate intermediate values
  - 2 x N values: vi,j=RSA(ki,m)
    - k0 = k_{MSB-1}, ..., k_l, 0
      k1 = k_{MSB-1}, ..., k_l, 1
- Calculate hypothetical power consumption
  - Calculate hi,j=HW (vi,j)
- Comparison: estimate correlation ri,j

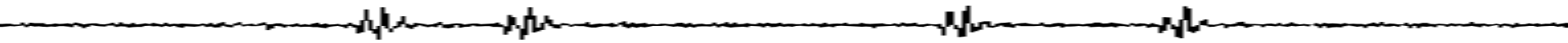
Case study for RSA: DPA using HW of result of multiplication (squaring)

Incorrect key guess

Correct key guess

© Messerges et al., CHES 1999

- Correct key hypothesis leads to DPA peak
- Time reveals when attacked intermediate value has been processed



Summary

- We looked at different power attacks on RSA
  - Simple vs. Differential power attacks (aka single observation vs. Many observations, or within trace analysis and across traces analysis), trace pair analysis
- A differential power attack compares real and simulated power values to derive information about the correct key hypothesis
  - This is done using distinguishers
    - Correlation analysis (using a suitable power model)
      - Is exactly the same principle as we did for differential timing analysis using a correlation distinguisher