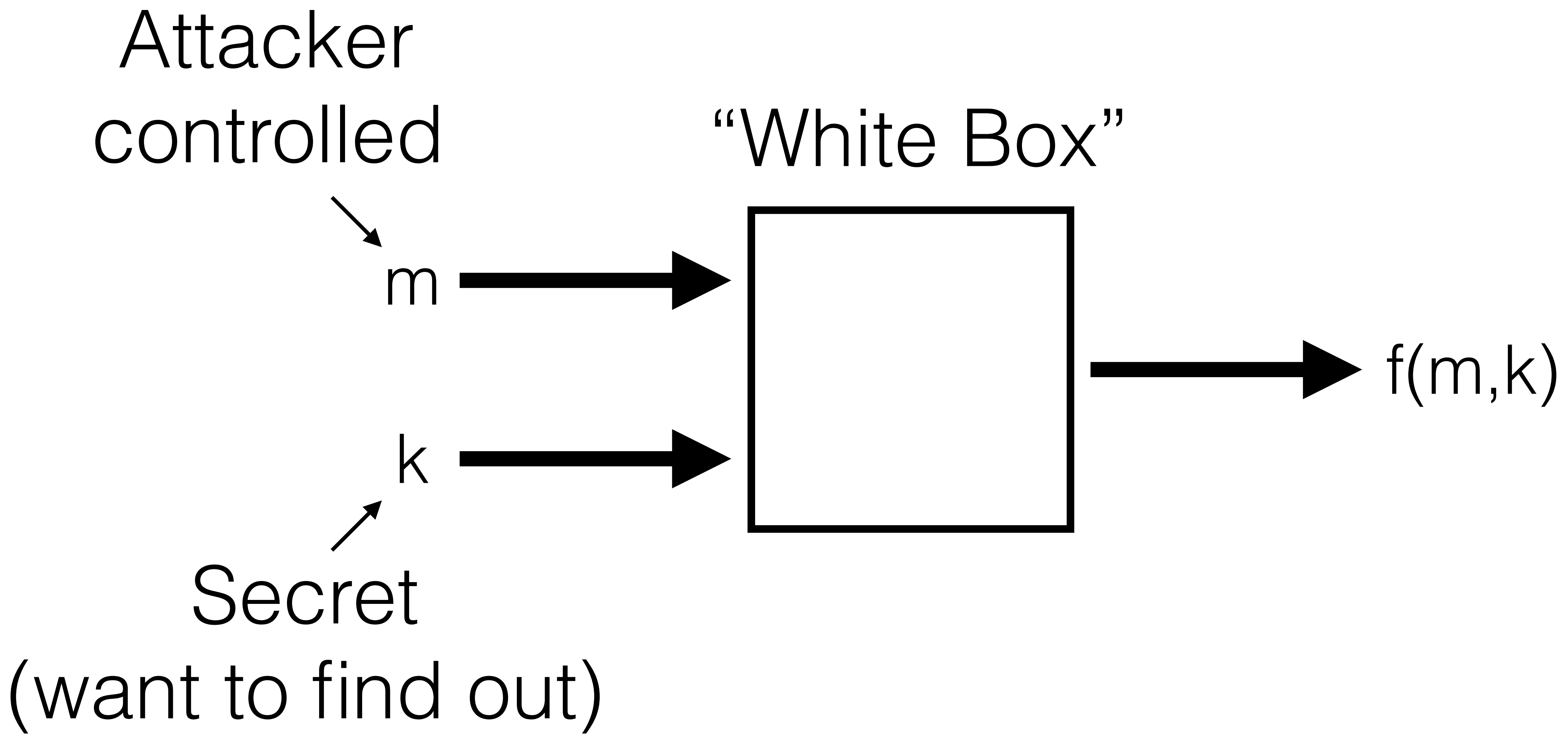
Attacking Hardware using Side Channel Power Analysis

Kevin Kiningham

Intro to Side Channels

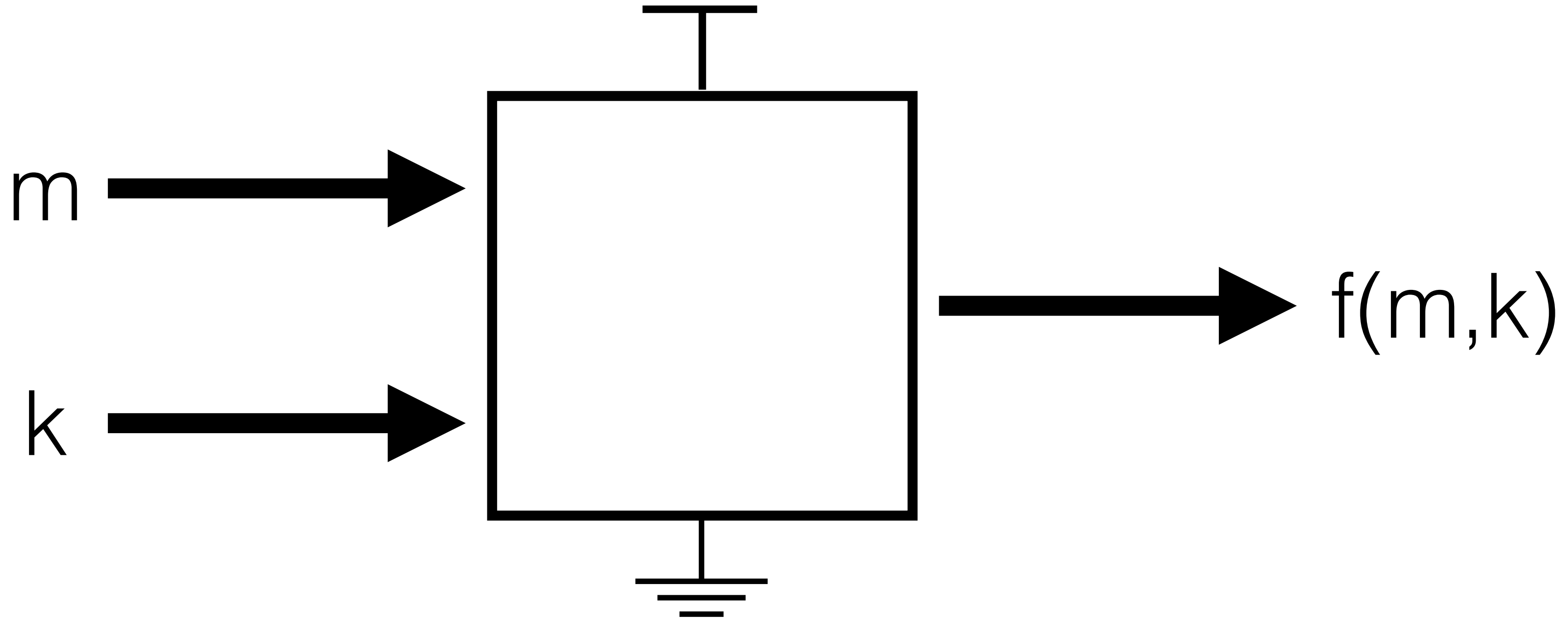
- Physical systems leak information while running
  - Power consumed
  - Time to compute
  - Electromagnetic radiation
  - etc…
- How can we use this information?

Intro to Side Channels



What is k?

Intro to Side Channels



Manual tells us:

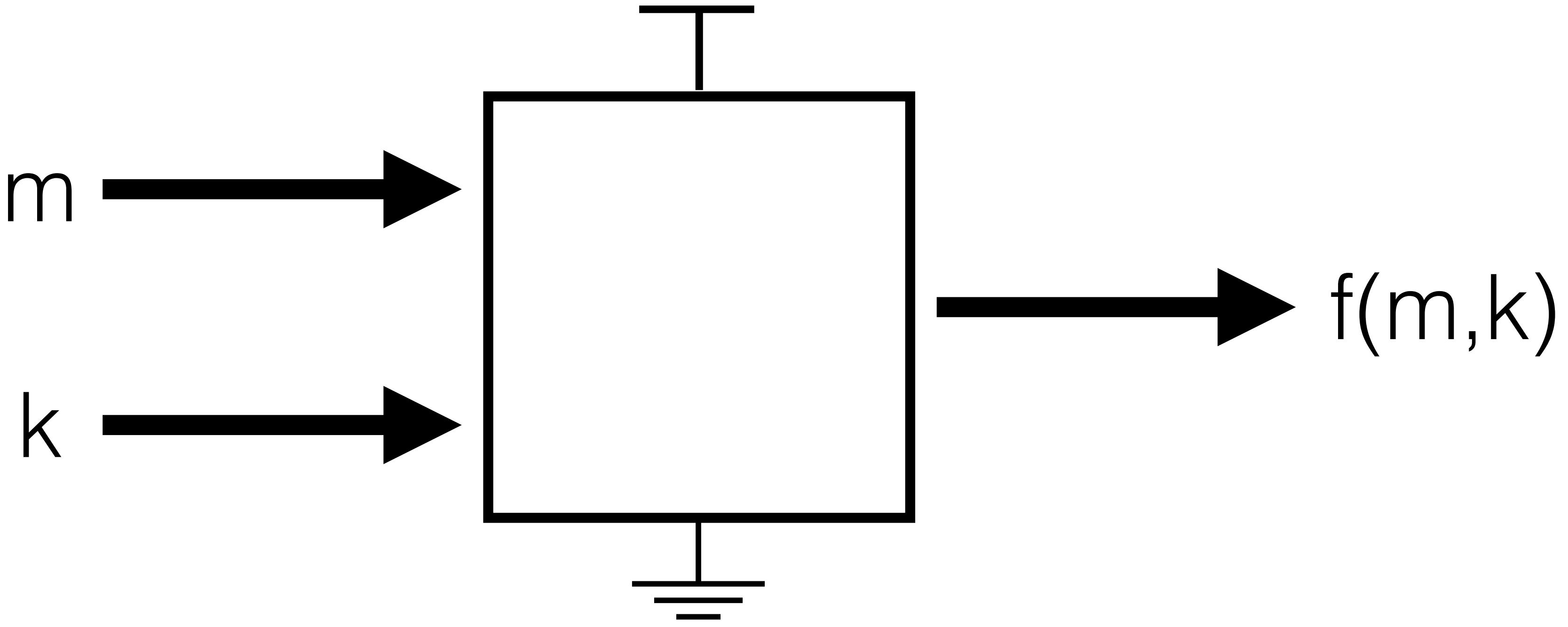
m	k	Energy Consumed
0	0	0 pJ
0	1	1 pJ
1	0	1 pJ
1	1	2 pJ

Measure:

m	k	Energy Consumed
0	???	0 pJ
1	???	1 pJ

What is k?

Intro to Side Channels



Manual tells us:

m	k	Energy Consumed
0	0	0 pJ
0	1	1 pJ
1	0	1 pJ
1	1	2 pJ

Measure:

m	k	Energy Consumed
0	0	0 pJ
1	0	1 pJ

What is k? k = 0

Intro to Side Channels

- Core Idea: Relate **leaked information** to **secret inputs**

- Allows us to discover secrets without breaking crypto

- Process of relating secret inputs to leaked information is called “Side Channel Analysis” (SCA)

Real World SCA

- Previous example made three major simplifications:

  1. Don’t have a table mapping inputs to power

  2. Energy consumption is stochastic (non-deterministic for a given input)

  3. Energy consumption varies over time (not a single value)

Power Model

- Problem #1: We don’t know power consumption for each possible inputs

- Solution: Assume power consumption follows a simple model
  - Ex: “Power consumption is **linear** with the **Hamming Weight** of the **output** of the circuit”

$$HW(f(k,m)))$$

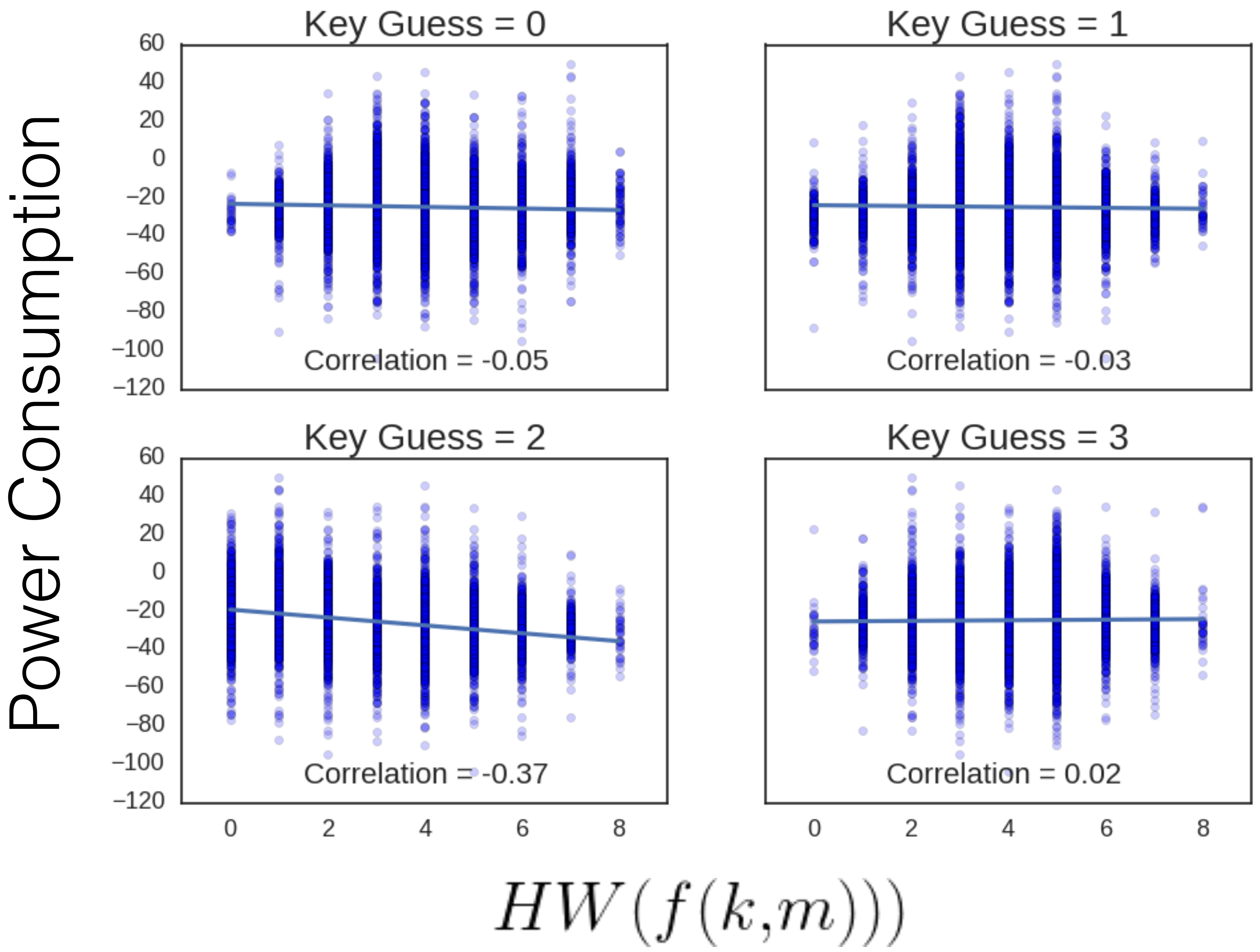
Power Model

- Problem #2: Our power model relies on the secret inputs
  - Recall: $HW(f(k,m)))$

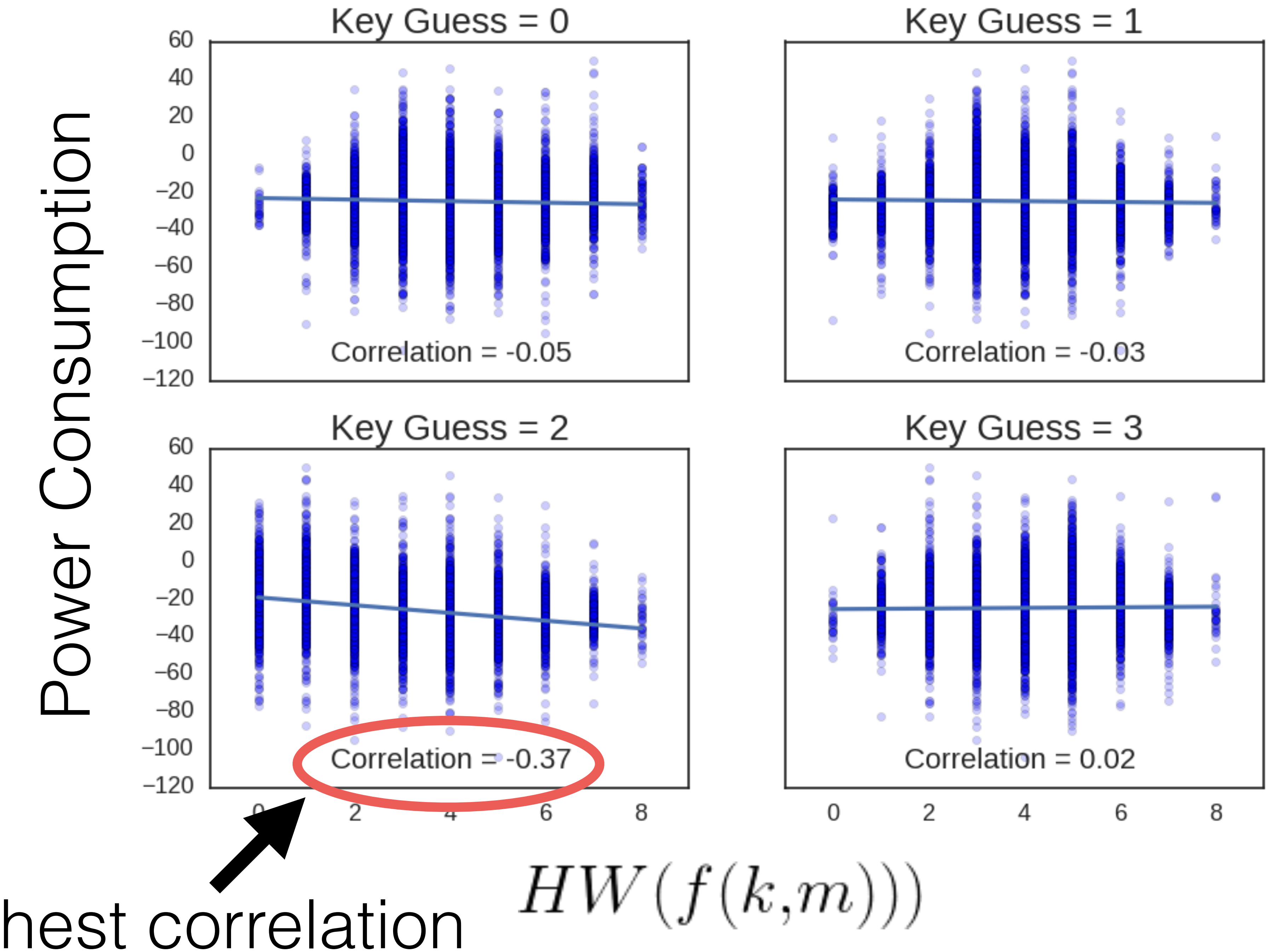
    k is unknown

- Solution: Try every possible value for the secret. Assume the value that best “matches” the actual power consumption is correct

Power Model


Key Guess = 0
Key Guess = 1
Key Guess = 2
Key Guess = 3
Correlation = -0.05
Correlation = -0.03
Correlation = -0.37
Correlation = 0.02
60
40
20
0
−20
−40
−60
−80
−100
−120
0
2
4
6
8
Power Consumption
$HW(f(k,m)))$

Power Model

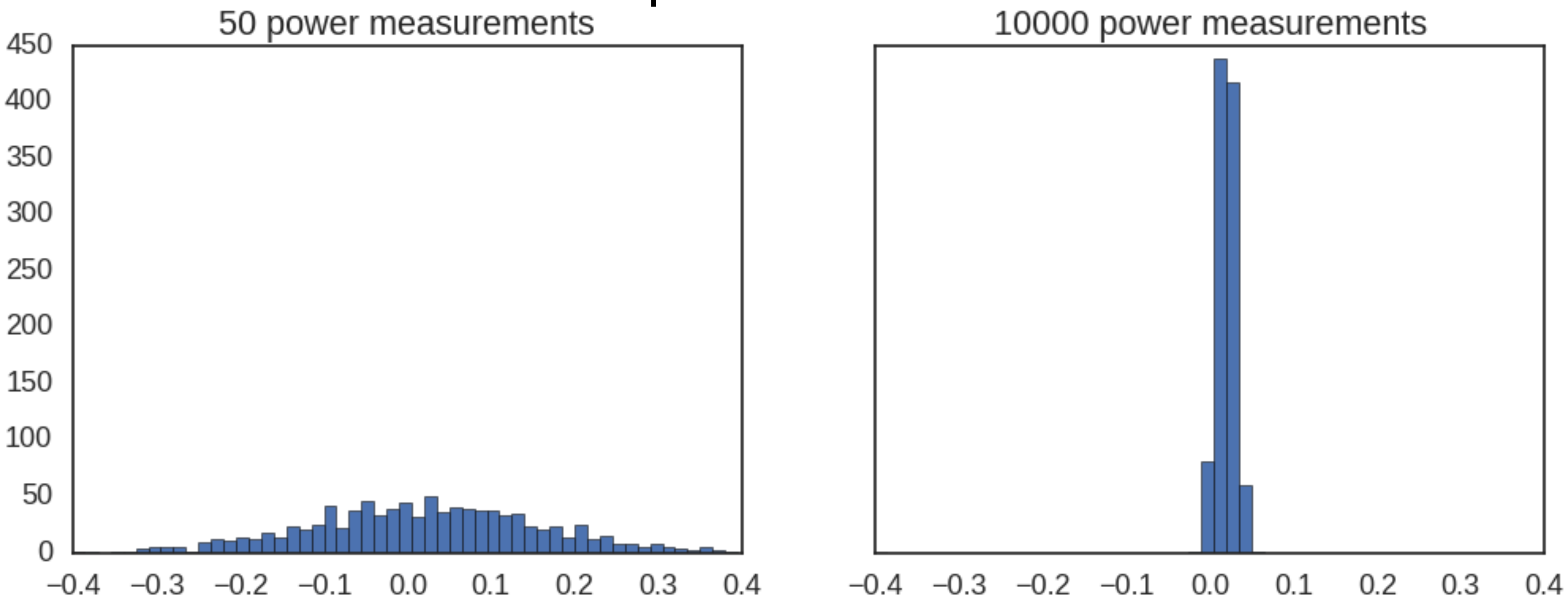


Highest correlation
gives correct Key

Power Model With Noise

- In real systems power measurements have lots of noise
  - Noise can be much larger than signal

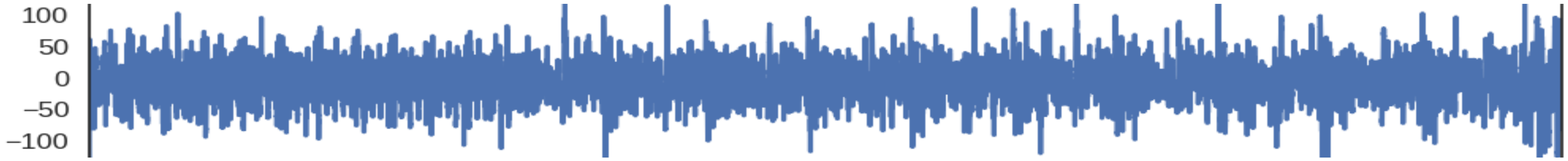
- Solution: Take lots of power measurements



Correlation Distribution

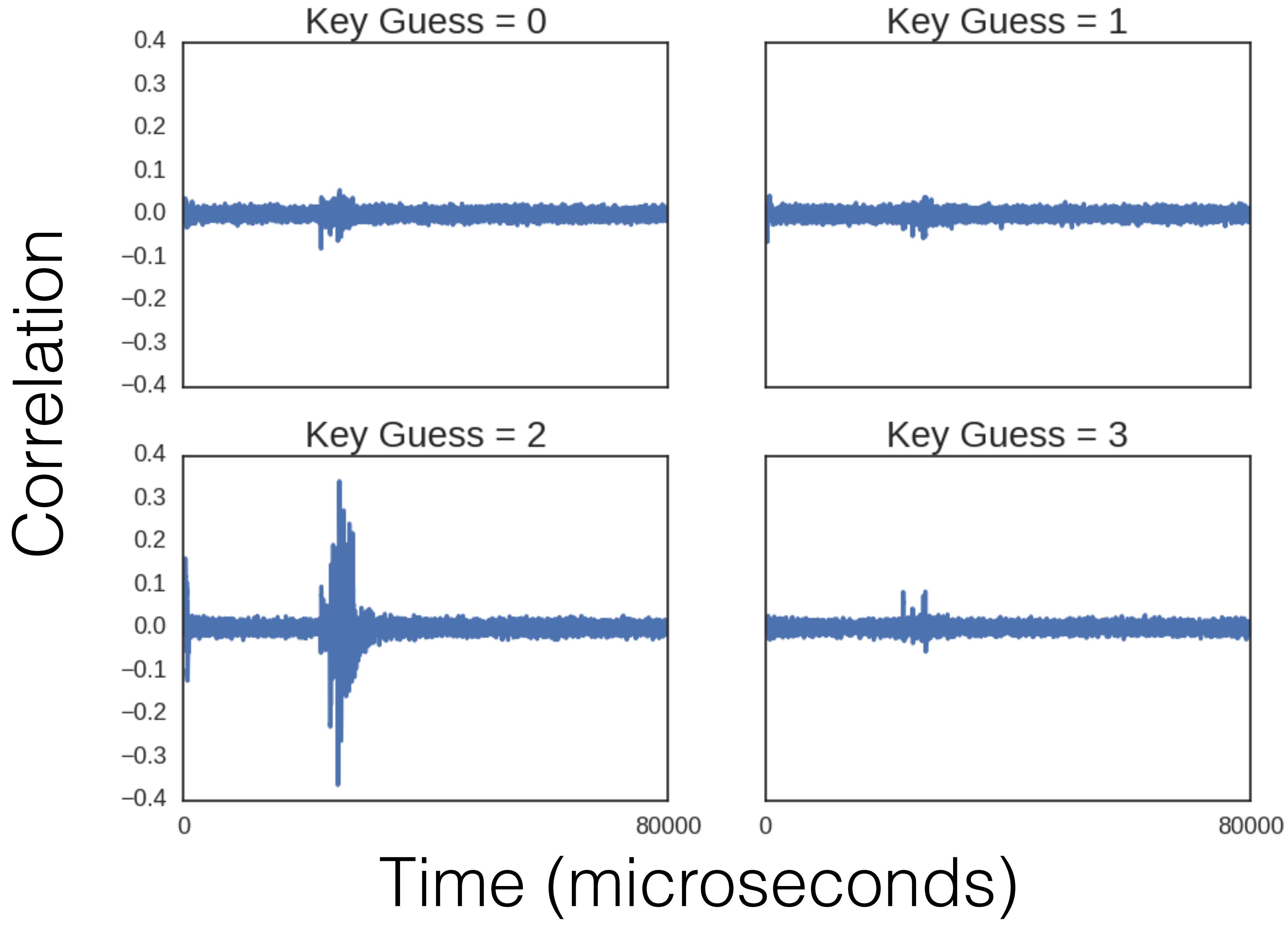
Time Varying Signal

- Power consumption changes over time
  - Not clear when targeted computation happens

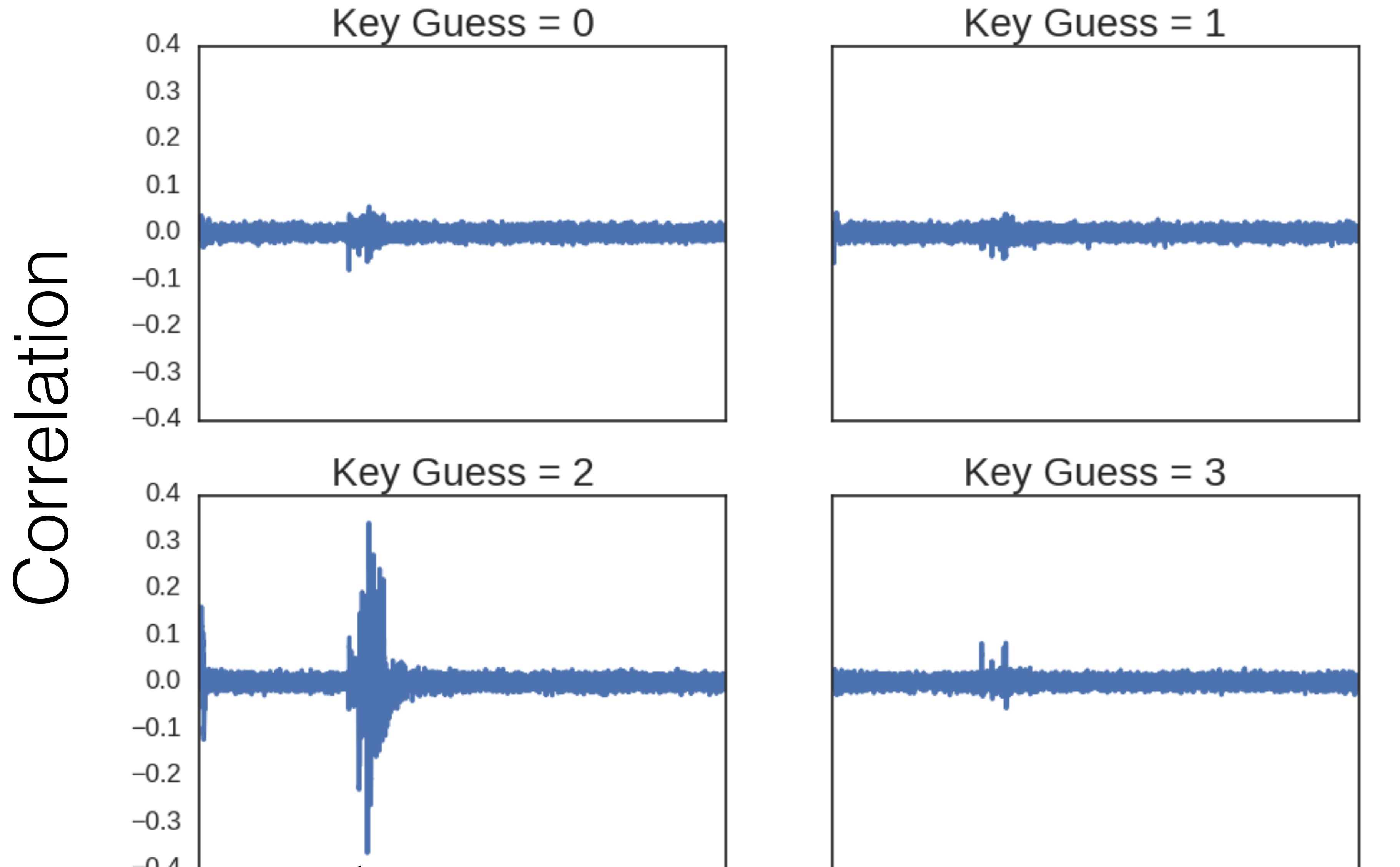


- Solution: Run the attack at each point in a trace and pick the point that correlates the best with the power model

Time Varying Signal



Time Varying Signal



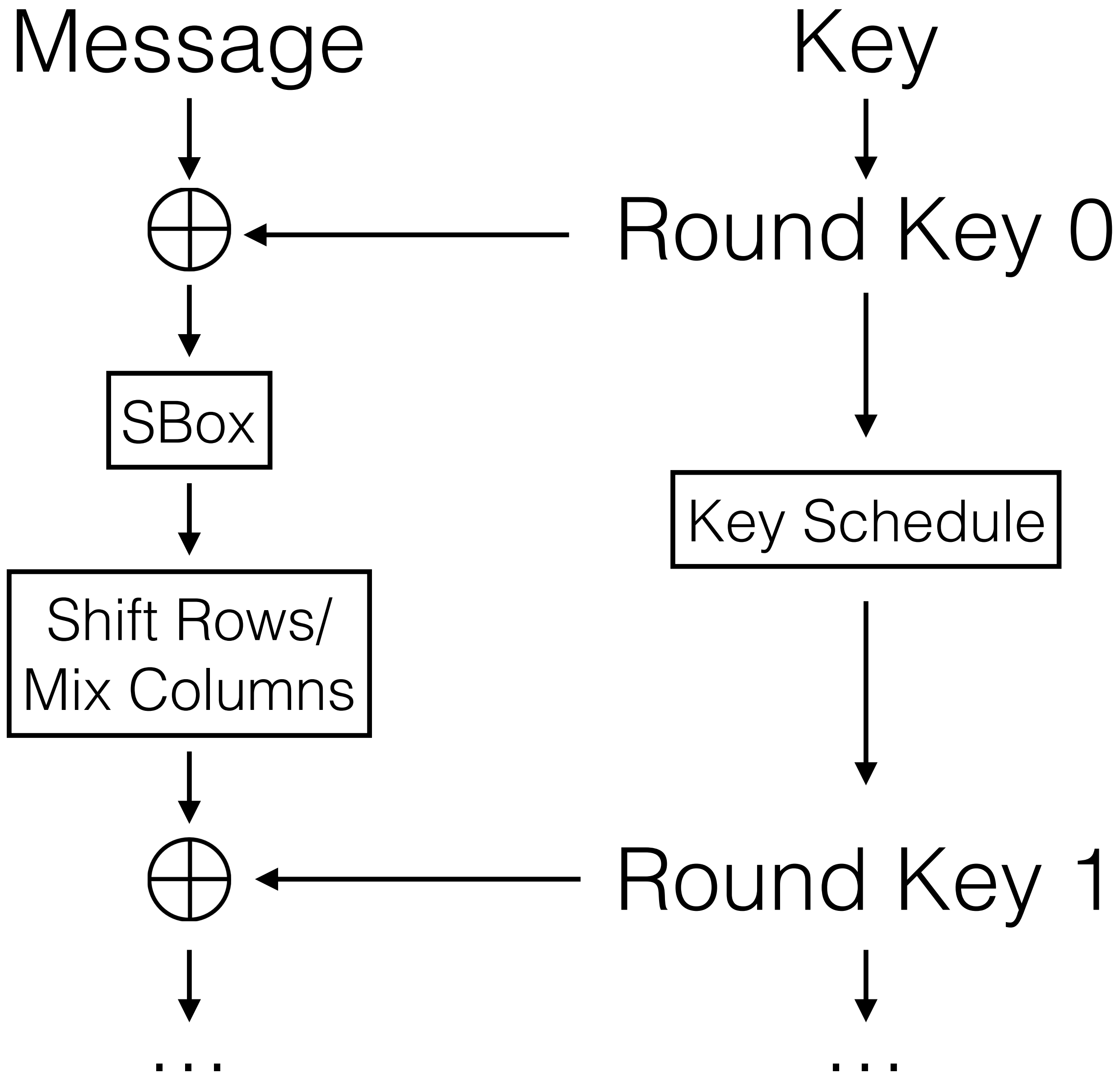
Highest correlation gives correct Key

Correlation Power Analysis (CPA)

- For every time period t:
  - For every key guess k:
    - Calculate the correlation between the power model and the observed power

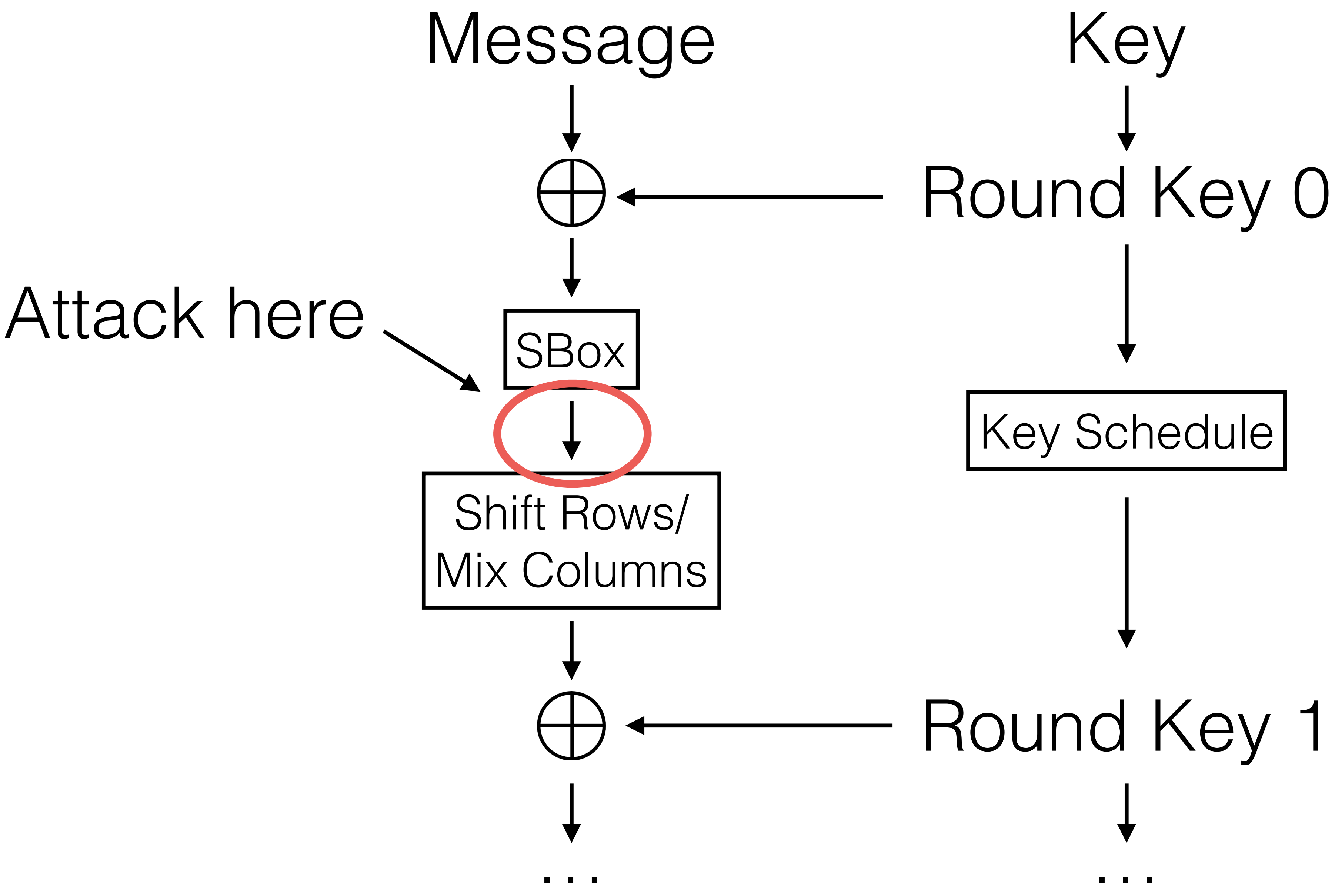
- Pick the key guess that maximizes the correlation across all time periods

Attacking AES128



Repeat 10 times

Attacking AES128



Message

Key

Round Key 0

Attack here

SBox

Key Schedule

Shift Rows/
Mix Columns

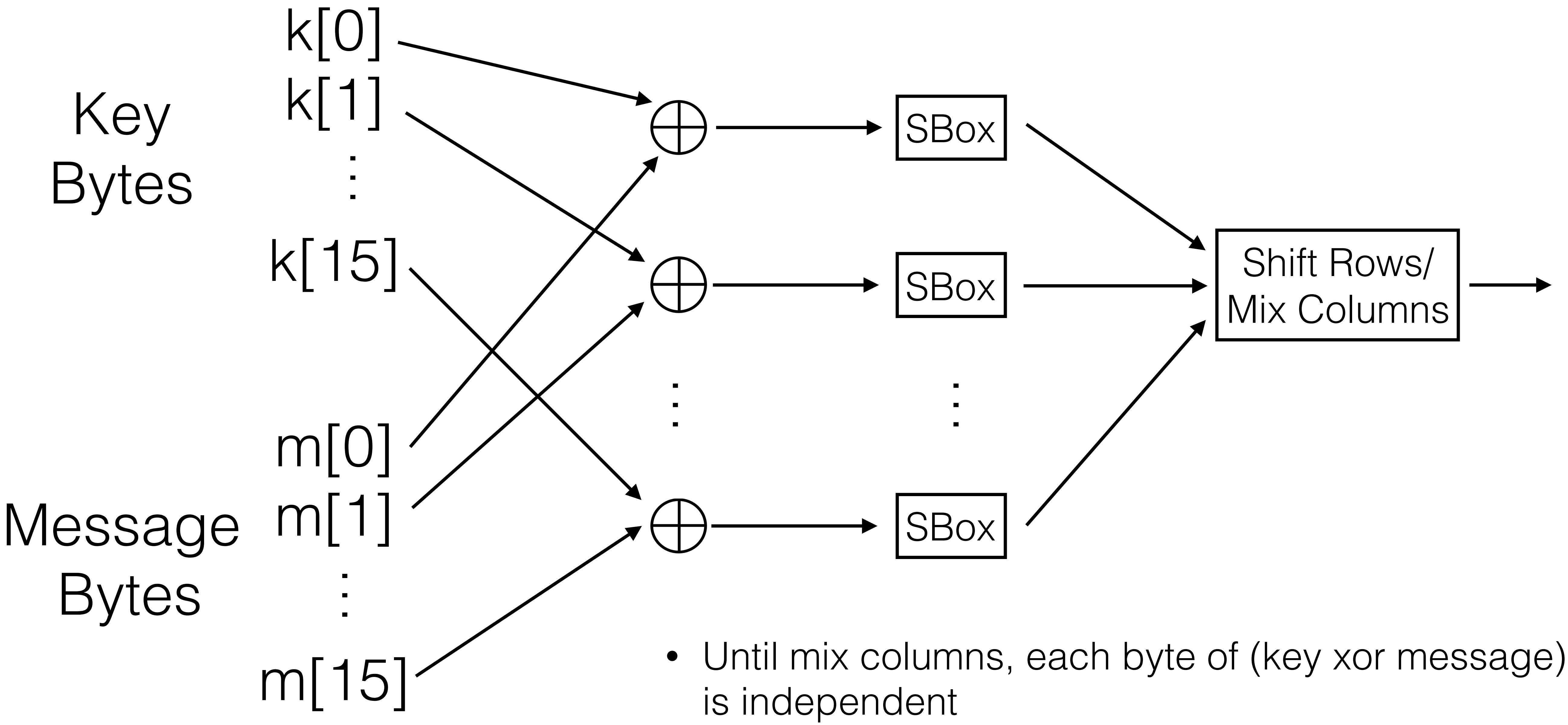
Round Key 1

…

…

Repeat 10 times

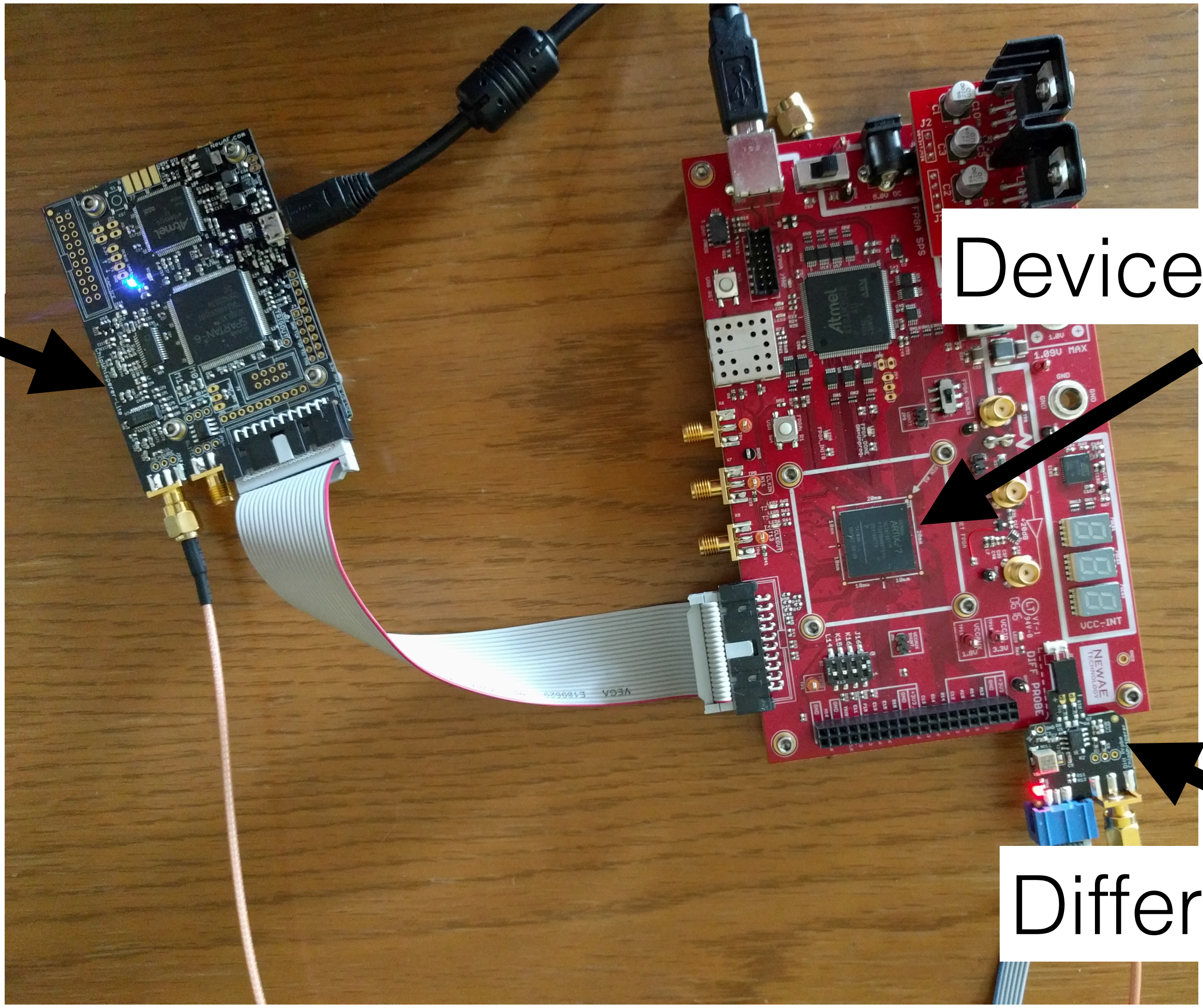
Attacking AES128



- Until mix columns, each byte of (key xor message) is independent
  - We can guess each byte of the key separately!

- Use HW(SBox(k[i] xor m[i])) as our power model

Running the Attack



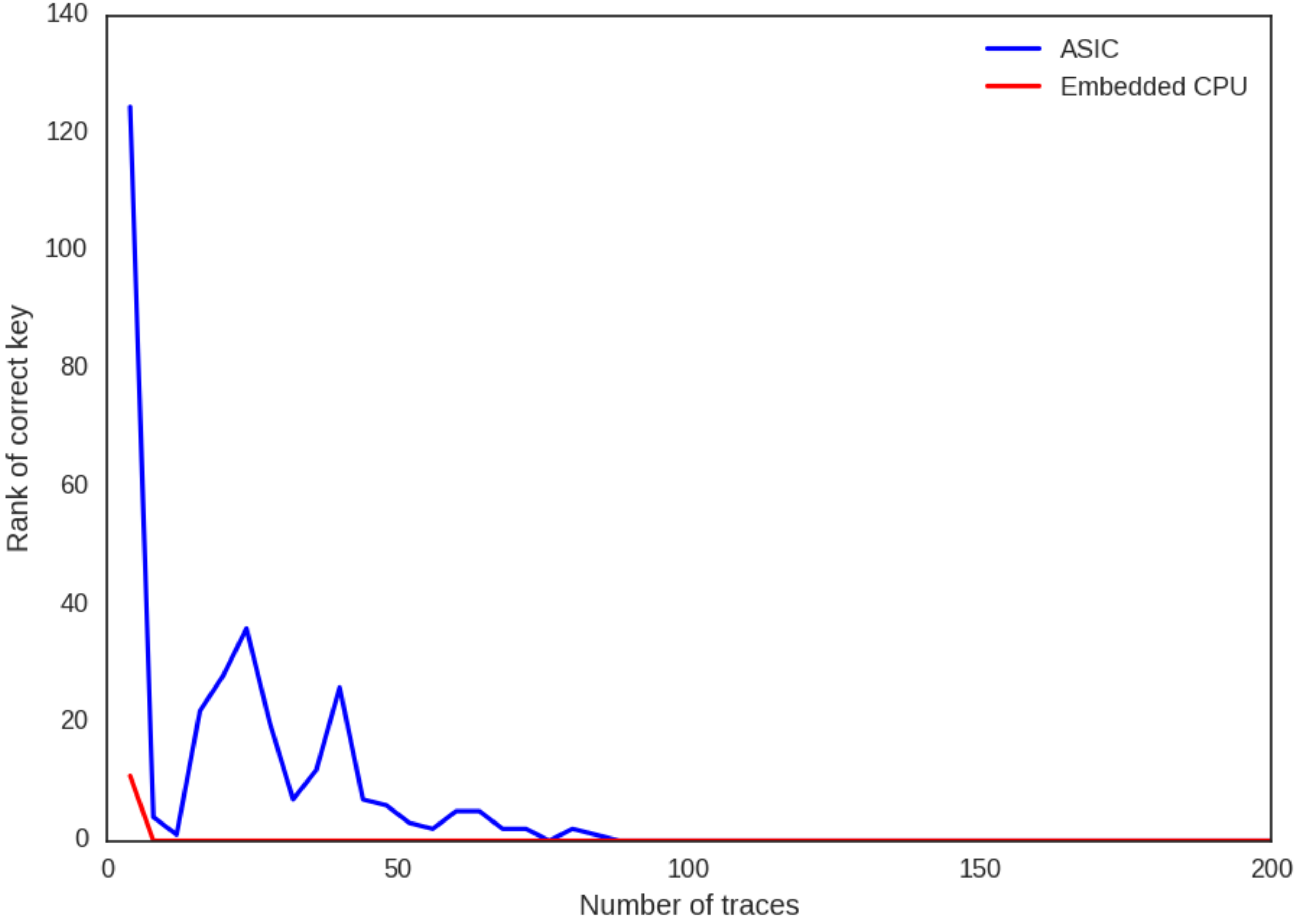
ADC + FPGA for sampling

Device Under Test

Differential Probe

Total Cost: ~$800

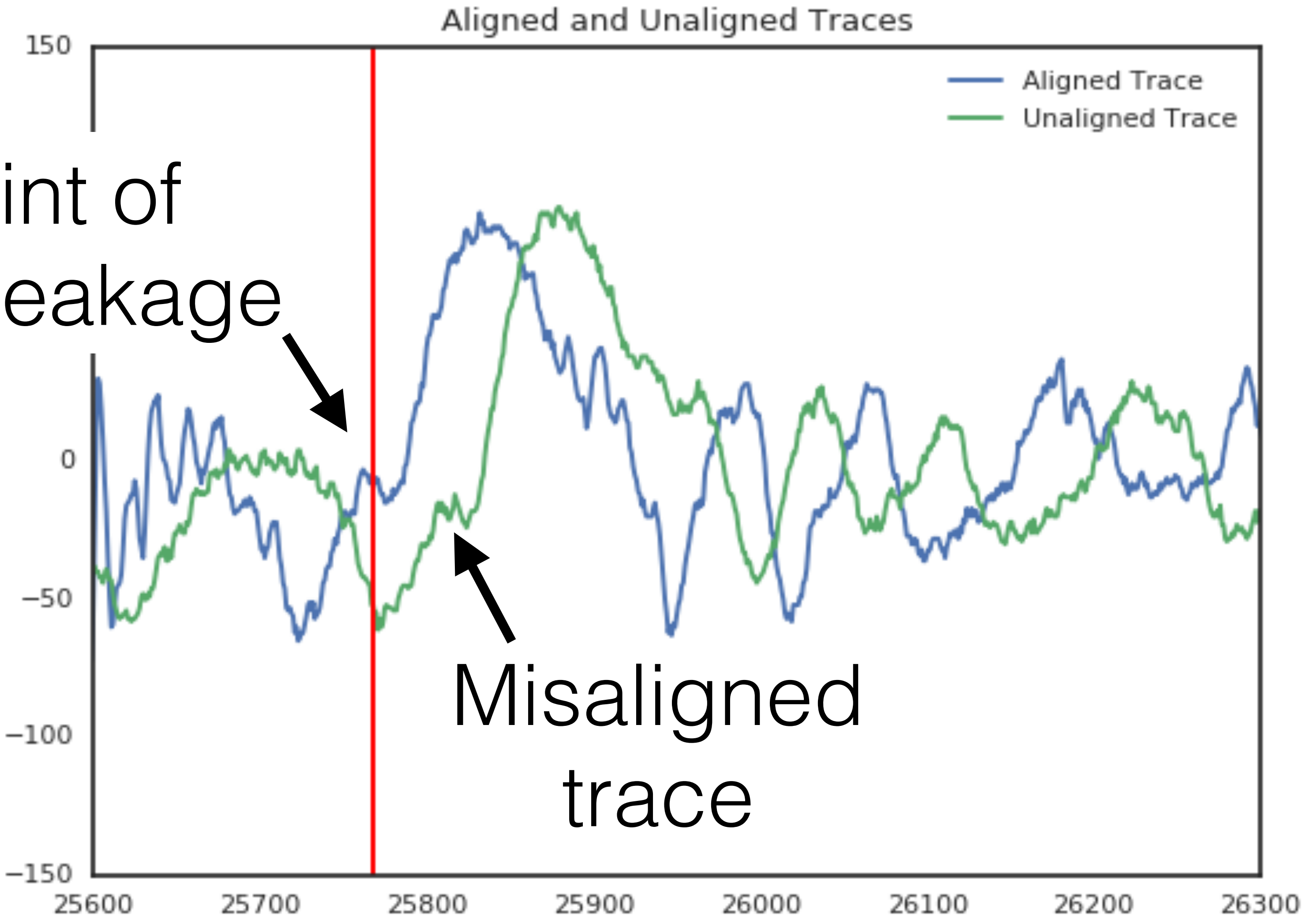
Results


ASIC
Embedded CPU
Rank of correct key
140
120
100
80
60
40
20
0
0
50
100
150
200
Number of traces

Countermeasures

- What kind of countermeasures are there?
  - Reduce signal
    - Use quieter circuits, add filtering
  - Adding Noise
  - Masking
    - Use cryptographic techniques to remove operations that operate directly on key (e.g. RSA blinding)
  - Variable timing
    - Reorder operations, insert dummy operations, variable frequency clock, etc

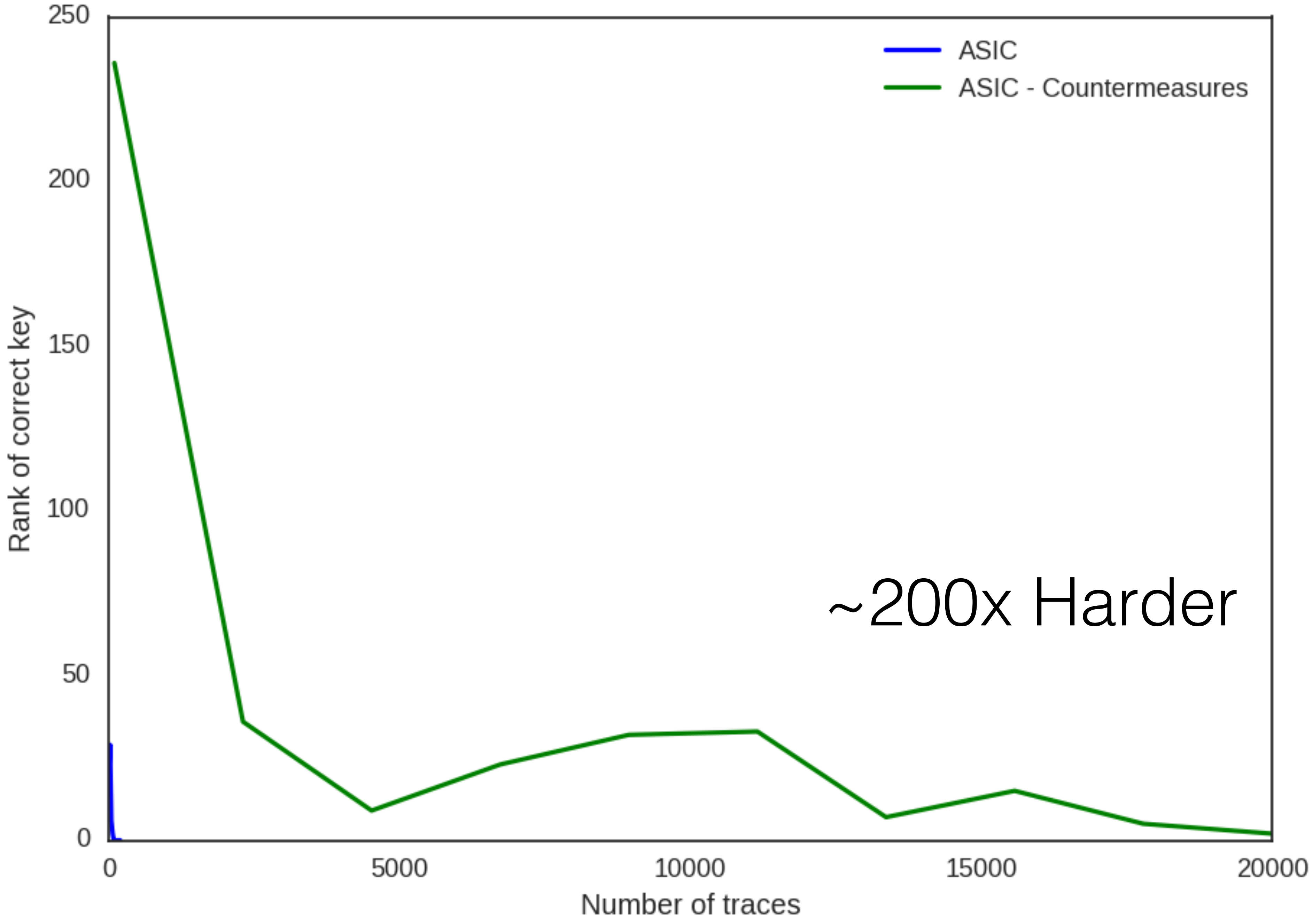
Variable Timing



Point of max leakage

Misaligned trace

Re-running with countermeasures



~200x Harder

Data

- https://github.com/google/power-traces