Side-Channel Attacks

Yu Bi

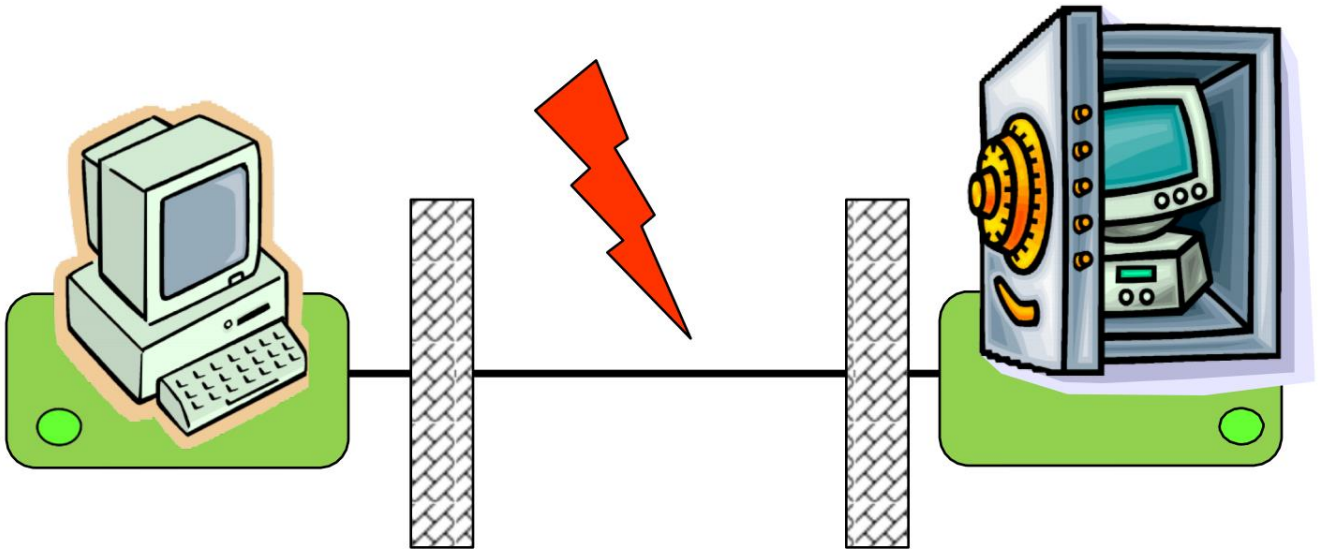
ELE594 – Special Topic on Hardware Security & Trust

University of Rhode Island

Introduction

- Classic cryptography views the secure problems with mathematical abstractions
- The classic cryptanalysis has had a great success and promise
  - Analyzing and quantifying crypto algorithms' resilience against attacks
- Recently, many of the security protocols have been attacked through physical attacks
  - Exploit weaknesses in the cryptographic system hardware implementation aimed to recover the secret parameters

Traditional Model



- Attack on channel between communicating parties
- Encryption and cryptographic operations in **black boxes**
- Protection by strong mathematic algorithms and protocols
- Computationally secure

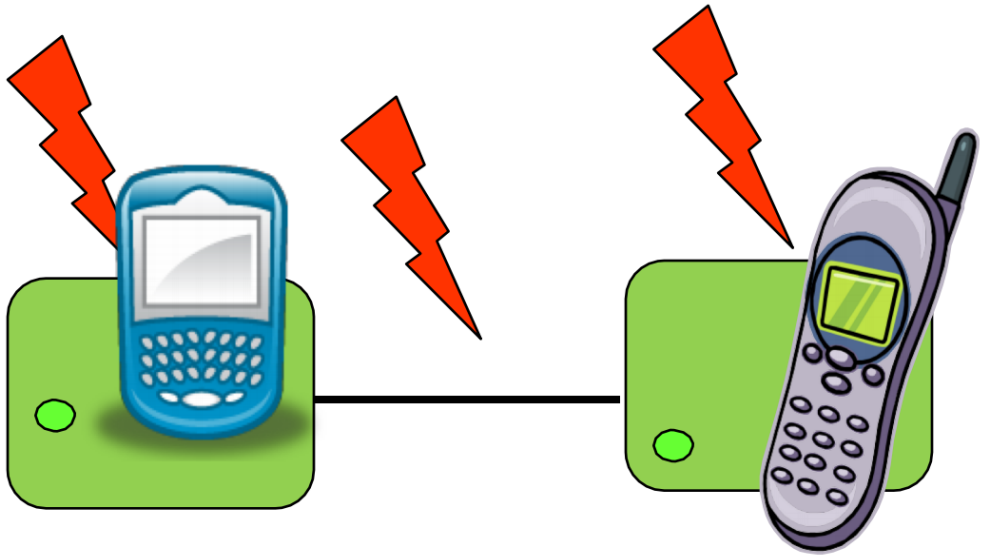
Cryptographic Devices

- A *cryptographic device* is an electronic device that implements a cryptographic algorithm and stores a cryptographic key. It is capable of performing cryptographic operations using that key.

IDENTIFICATION PAYMENT ***COMMUNICATION*** ***MULTIMEDIA***

- *Embedded*: it is exposed to adversaries in a hostile environment; full physical access, no time constraints
  - Remark: the adversary might be a legitimate user!

Embedded Security Affected



- New Model (also simplified view):
  - Attack on channel and endpoints
  - Encryption and cryptographic operations in **gray boxes**
  - Protection by strong mathematic algorithms and protocols
  - **Protection by secure implementation**

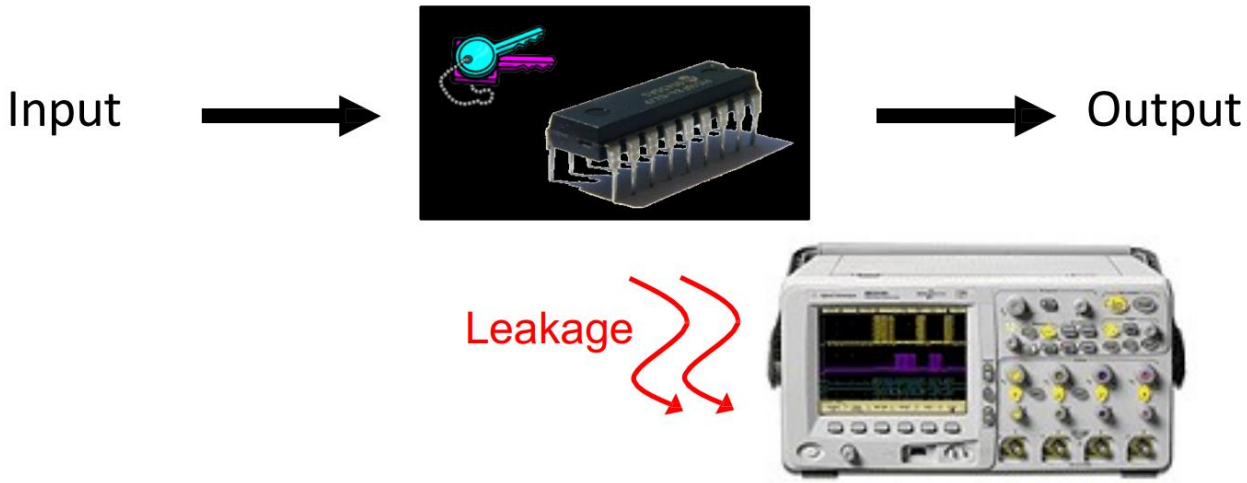
- ***Need secure implementations not only algorithms***

Side-Channel Leakage

Physical attacks ≠ Cryptanalysis

(gray box, physics) (black box, maths)

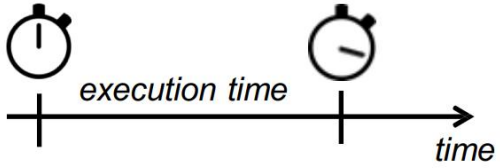
- Does not tackle the algorithm's math



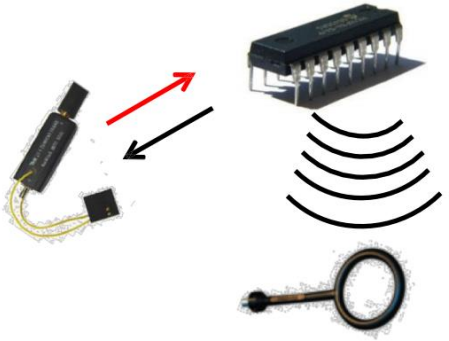
- Observe physical quantities in the device's vincinity and use additional information during cryptanalysis

Side-Channel Leakage

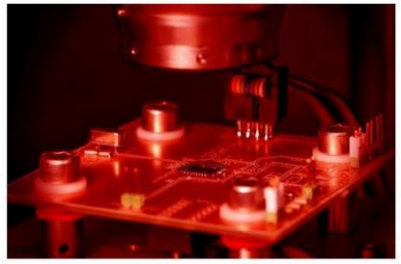
- Passive:
  - Timing
    - Overall or “local” execution time



  - Power, Electromagnetic (EM) radiation
    - Predominant CMOS technology
    - Dynamic power consumption
    - Electric current induces an EM field



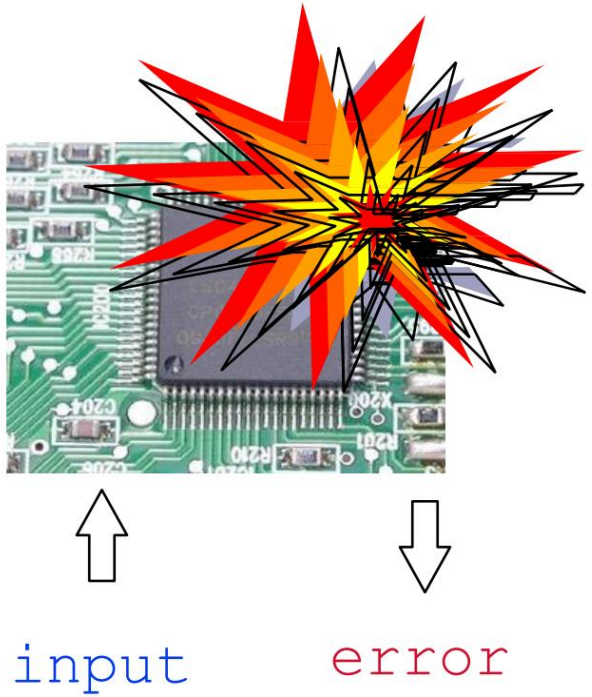
  - More exotic but shown to be practical
    - Sound, temperature, …



- Invasive: Photonic emissions

Fault-injection Attacks

- Non-(semi)invasive: apply combination of unaccounted environmental conditions
  - Vcc
  - Glitch
  - Clock
  - Temperature
  - UV
  - Light
  - X-Rays
  - …



- And bypass security mechanisms or infer secrets

Fault-injection Attacks

- Invasive: exploit faulty behavior provoked by physical stress applied to the device

  - Laser fault injection allows to target a relatively small surface area of the target device
  - Laser pulse frequency ~ 50Hz
  - Fully automated scan of chip surface

  - Once you have a weak spot: perturbate and exploit

Side-Channel Summary

- **Power Consumption** -- Logic circuits typically consume differing amounts of power based on their input data.
- **Electro-Magnetic** -- EM emissions, particularly via near-field inductive and capacitive coupling, can also modulate other signals on the die.
- **Optical** -- The optical properties of silicon can be modulated by altering the voltage or current in the silicon.
- **Timing and Delay** -- Timing attacks exploit data-dependent differences in calculation time in cryptographic algorithms.
- **Acoustic** -- The acoustic emissions are the result of the piezoelectric properties of ceramic capacitors for power supply filtering and AC to DC conversion.

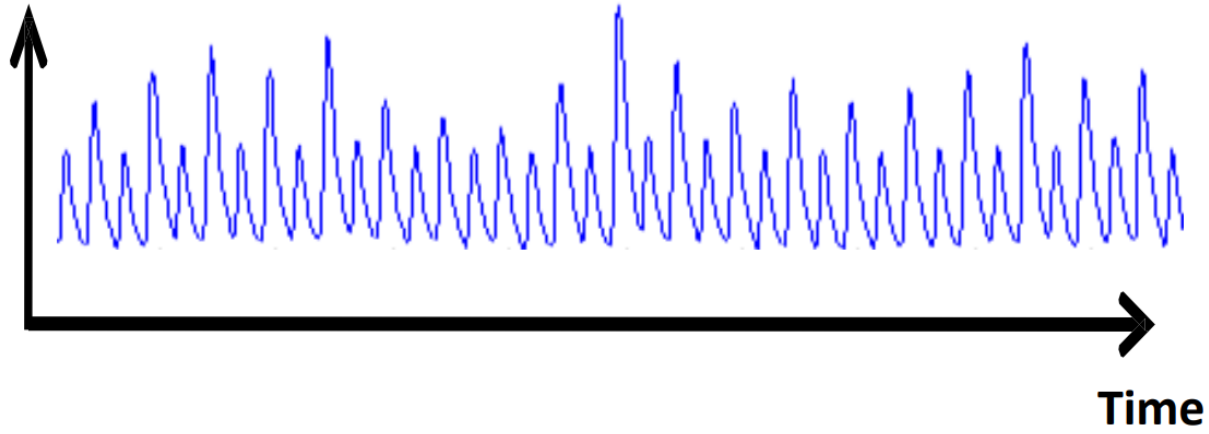
Side-Channel Attacks

- Side-Channel attacks aim at side-channel inputs and outputs, bypassing the theoretical strength of cryptographic algorithms

- Five commonly exploited side-channel emissions:
  - Power Consumption
  - Electro-Magnetic
  - Optical
  - Timing and Delay
  - Acoustic

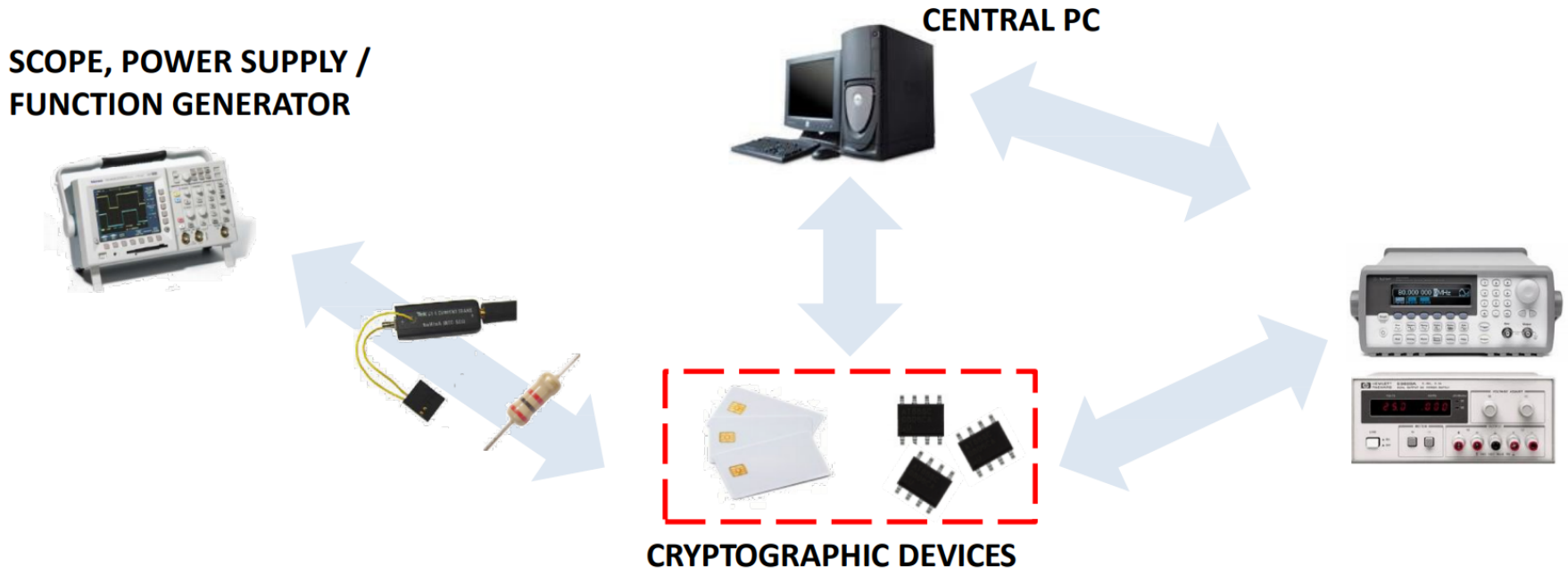
Power Consumption

- **Not average power over time, not peak power**
- Instantaneous power over time
  - Trace or curve, many samples



Measuring Power Consumption

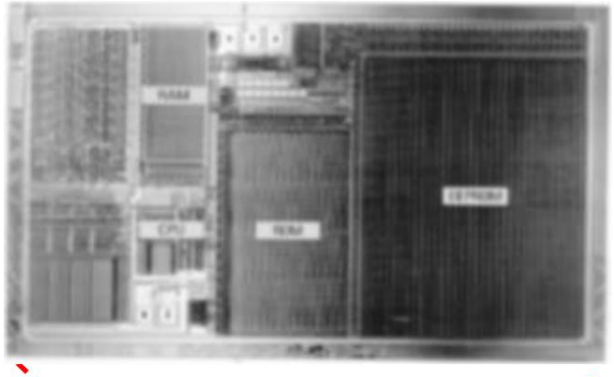
Typical (automated) measurement setup



Measuring Power Consumption

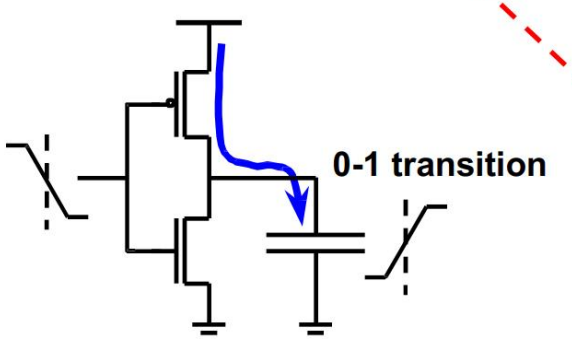
- **Logic**: constant supply voltage, supply current varies

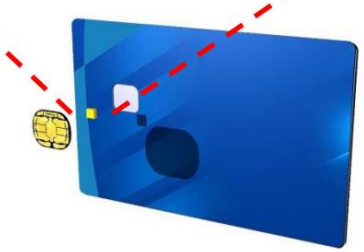
- **Predominant technology**: CMOS
  - Low static power consumption
  - Relatively high dynamic power consumption
  - Power consumption depends on input



- **CMOS inverter**:

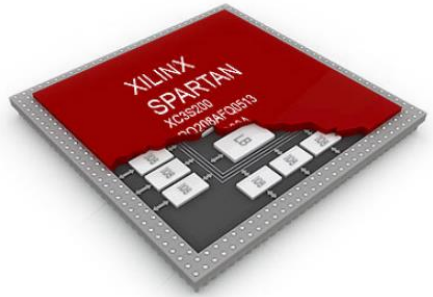
Input	Output	Current
0 → 0	1 → 1	Low
0 → 1	1 → 0	Discharge
1 → 0	0 → 1	Charge
1 → 1	0 → 0	Low





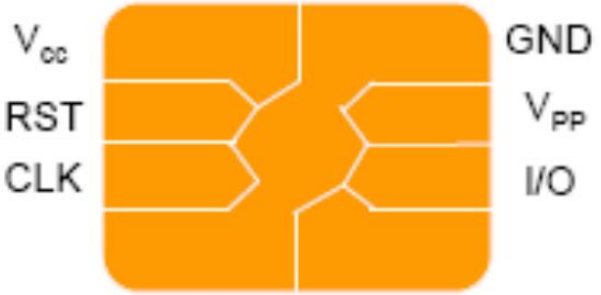
Hardware Targets

- Two common victims of hardware cryptanalysis are **smart cards** and **FPGAs**
  - Attacks on smart cards are applicable to any general purpose processor with a fixed bus architecture.
  - Attacks on FPGAs are also reported. FPGAs represent application specific devices with parallel computing opportunities.



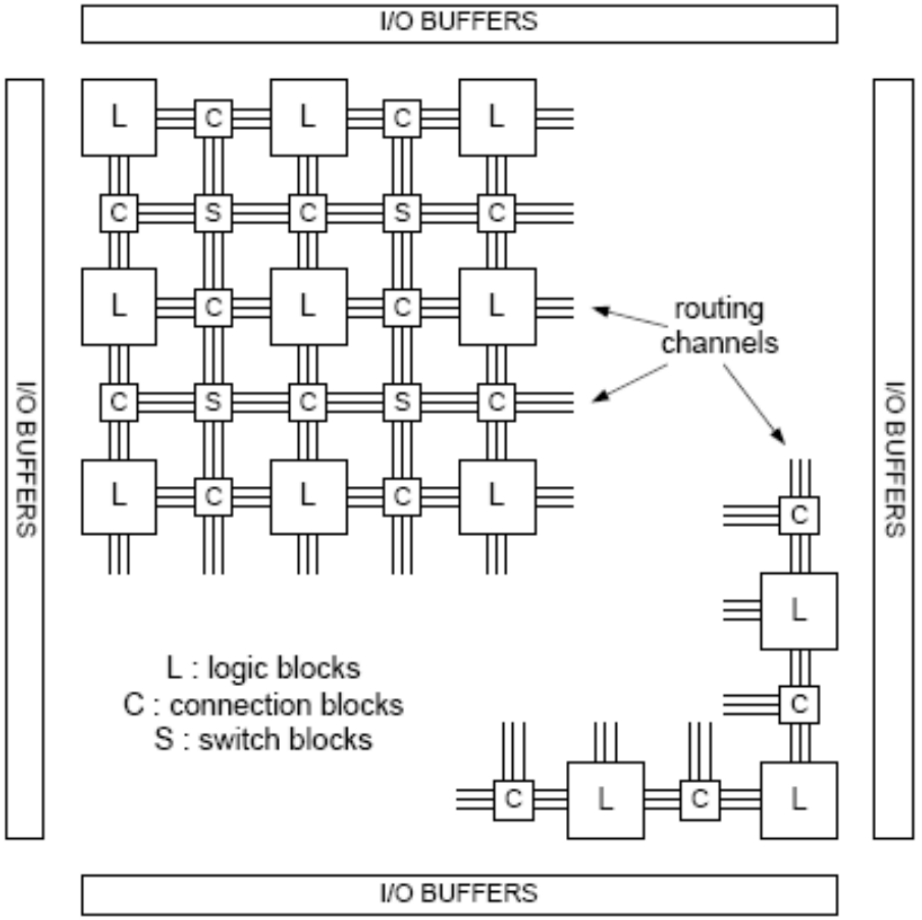
Smart Cards

- Smart cards have a small processor (8bit in general) with ROM, EEPROM and a small RAM
- **Eight wires** connect the processor to the outside world
- **Power supply**: There is no internal battery
- **Clock**: There is no internal clock
- Typically equipped with a **shield** that destroys the chip if a tampering happens



FPGA

- FPGAs allow parallel computing
- Multiple programmable configuration bits



Attack Model

- Consider a device capable of implementing the cryptographic function
- The key is usually stored in the device and protected
- Modern cryptography is based on Kerckhoffs's assumption → all of the data required to operate a chip is entirely hidden in the key

- ***Attacker only needs to extract the key***

Attack Steps

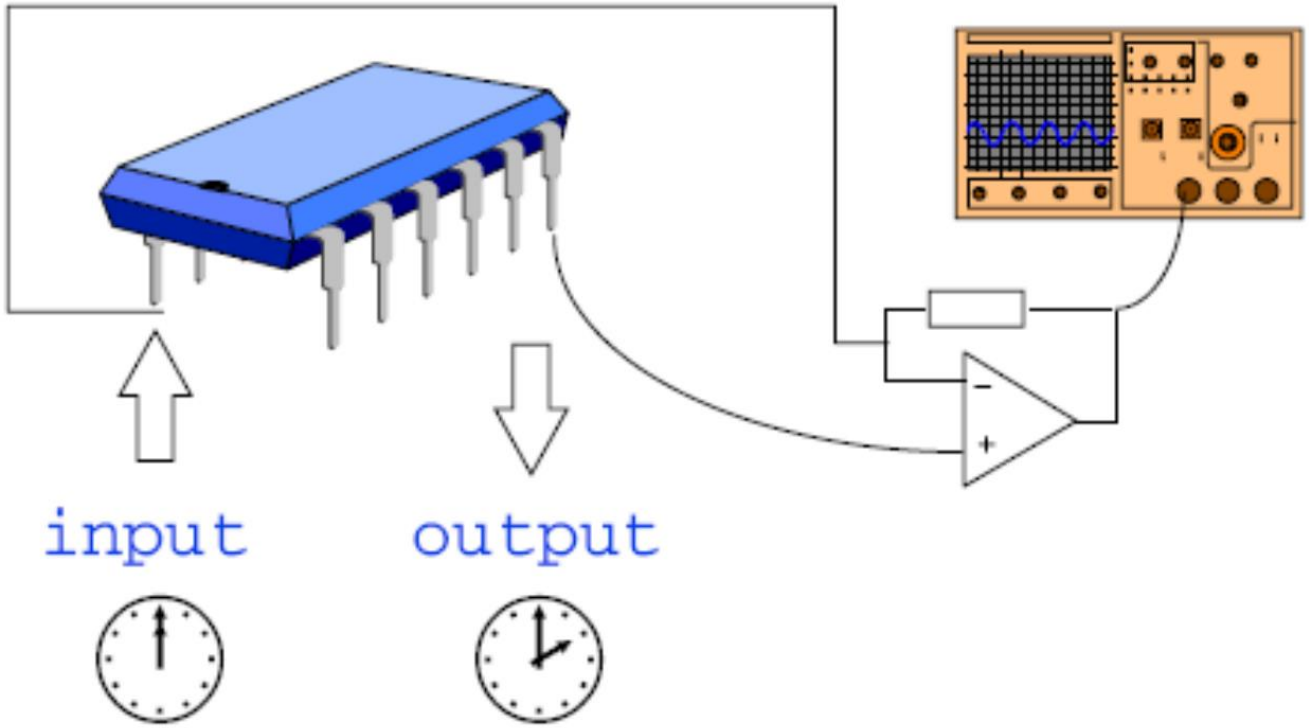
- Such attacks are usually composed of two phases:
  - **Interaction phase**: interact with the hardware system under attack and obtain the physical characteristics of the device
  - **Analysis phase**: analyze the gathered information to recover the key

Attack Classification

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

Power Attack

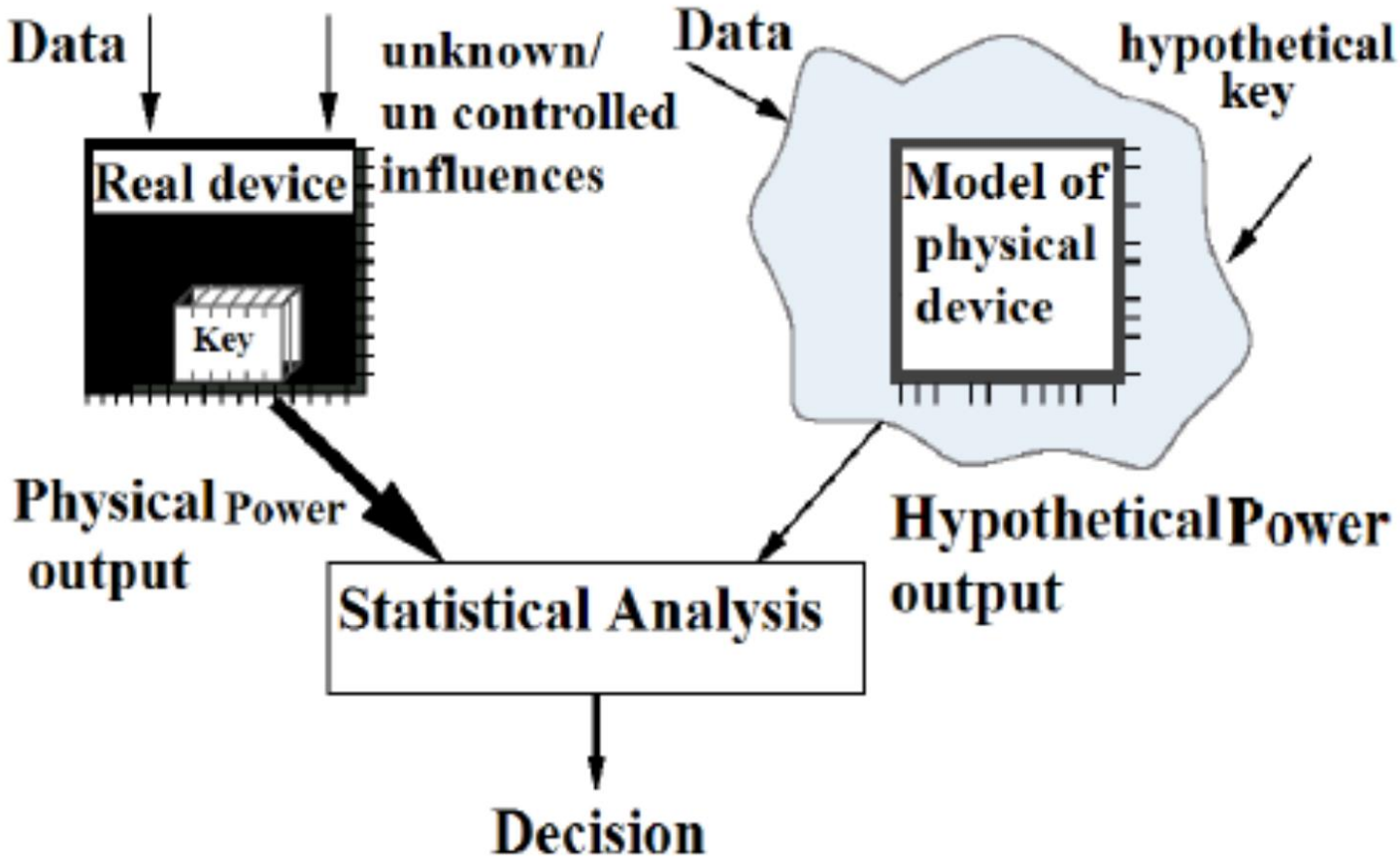
- Measure the circuit's processing time and current consumption to infer what is going on inside it.



Measuring Phases

- The task is usually straightforward
  - Easy for smart cards: the energy is provided by the terminal and the current can be read
- Relatively inexpensive (<$1000) equipment can digitally sample voltage differences at high rates (1GHz++) with less than 1% error
- Device's power consumption depends on many things, including its structure and data being processed

Power Attack Flowchart



Simple Power Analysis

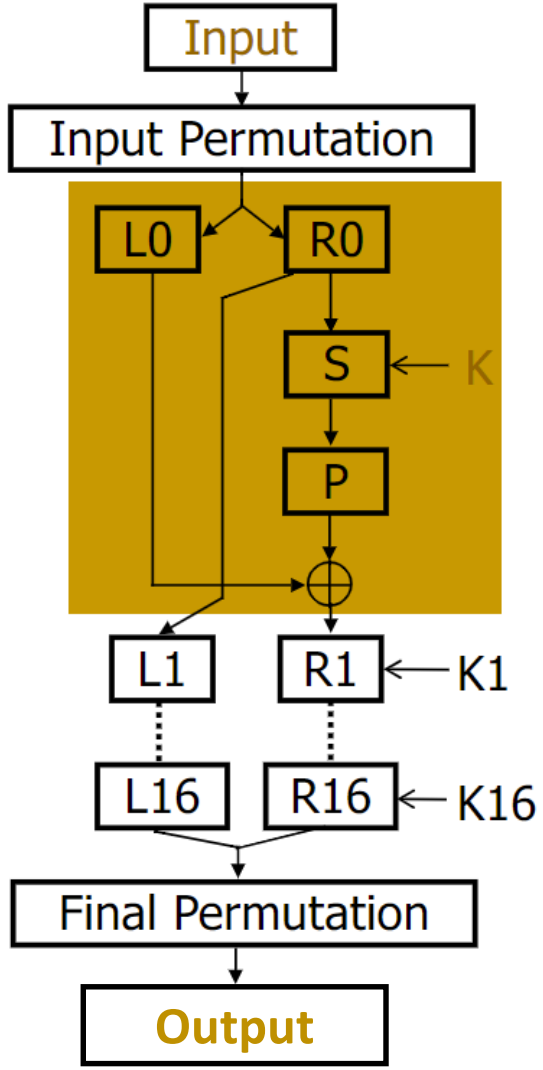
- Originally proposed by Paul Kocher, 1996
- Monitor the device's power consumption to deduce information about data and operation
- Example: SPA on DES – smart cards
  - The internal structure is shown on the next slide
- Summary of DES – a block cipher
  - a product cipher
  - 16 rounds iterations
    - substitutions (for confusion)
    - permutations (for diffusion)
  - Each round has a *round key*
    - Generated from the user-supplied key

Known

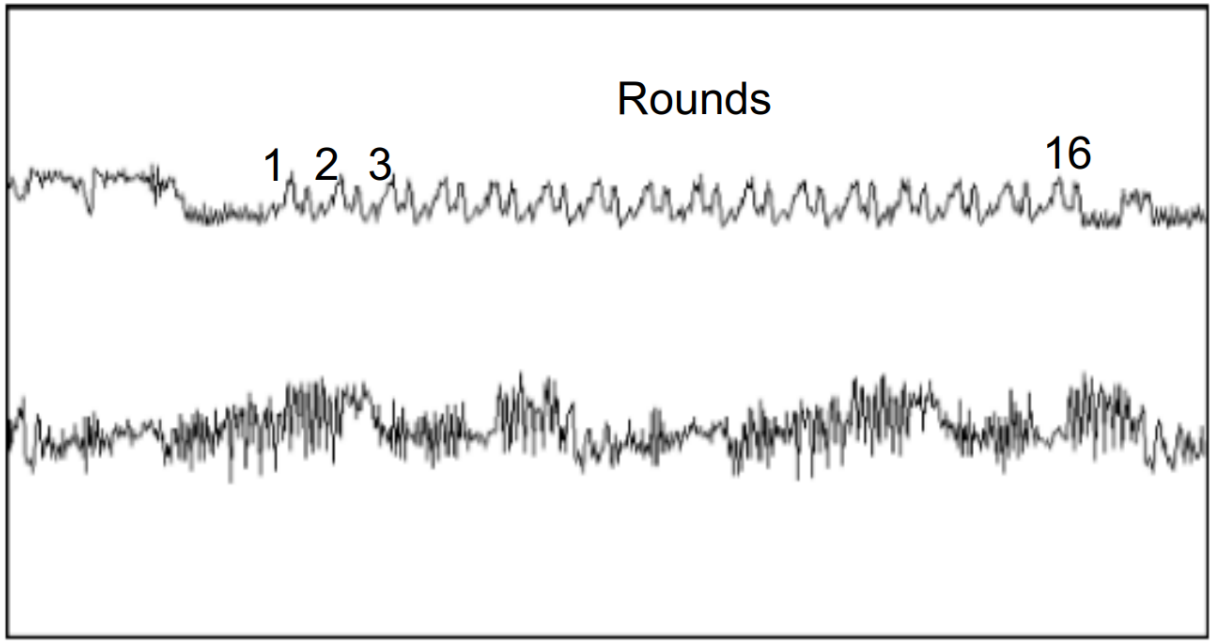
DES

- Input: 64 bits (a block)
- Li/Ri– left/right half of the input block for iteration i (32 bits) – subject to substitution S and permutation P
- K - user-supplied key
- Ki - round key:
  - 56 bits used +8 unused

    (unused for E but often used for error checking)
- Output: 64 bits (a block)
- Note: Ri becomes L(i+1)
- All basic op's are simple logical ops
  - Left shift / XOR

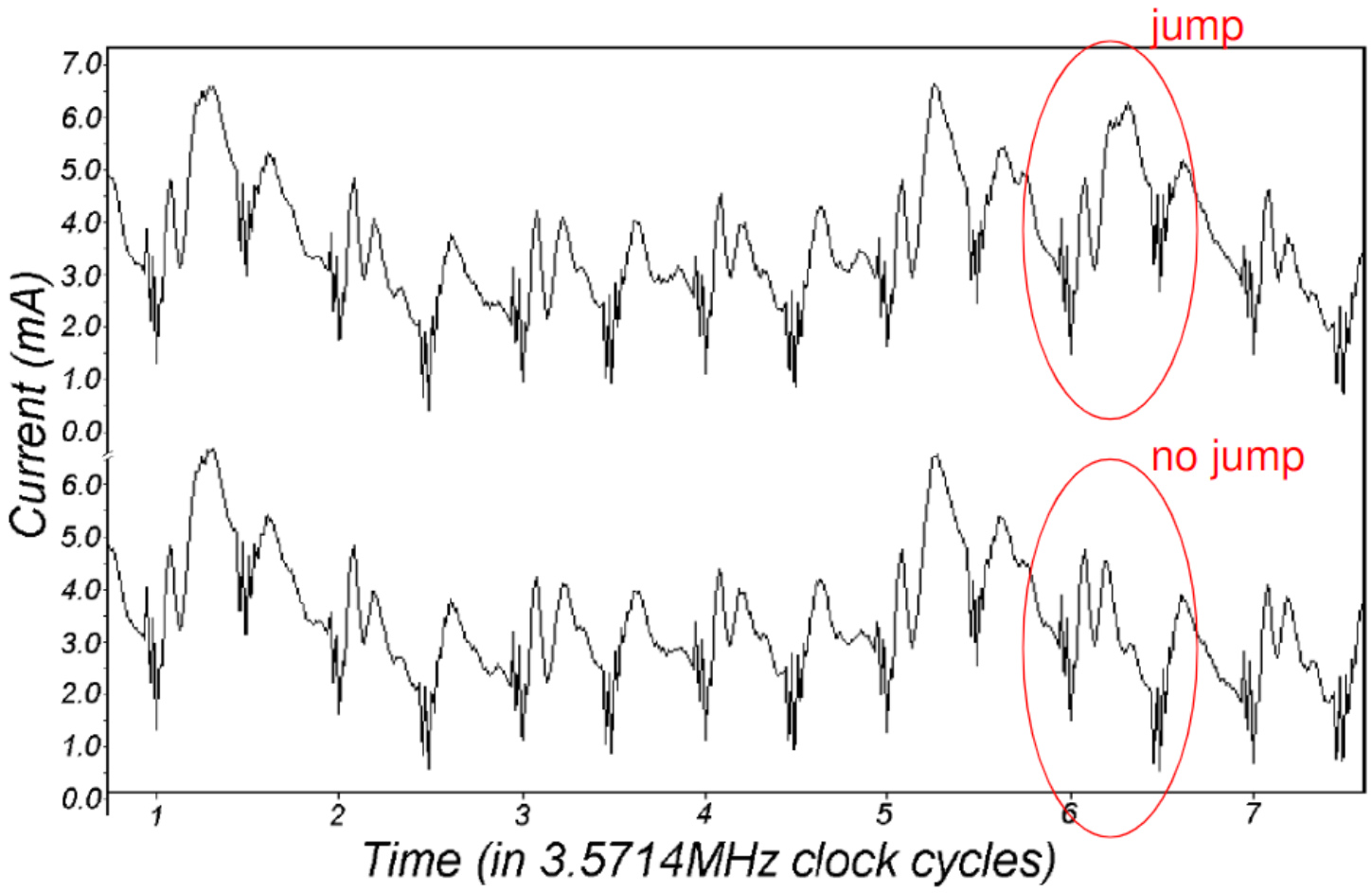


SPA on DES



- The upper trace – entire encryption, including the initial phase, 16 DES rounds, and the final permutation
- The lower trace – detailed view of the second and third rounds
- **The power trace can reveal the instruction sequence**

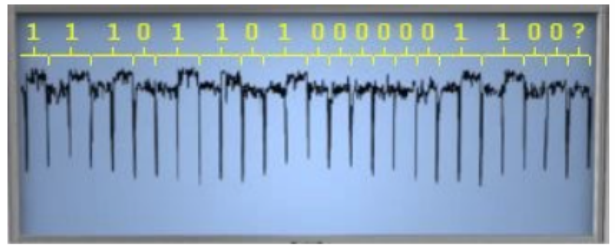
SPA on DES

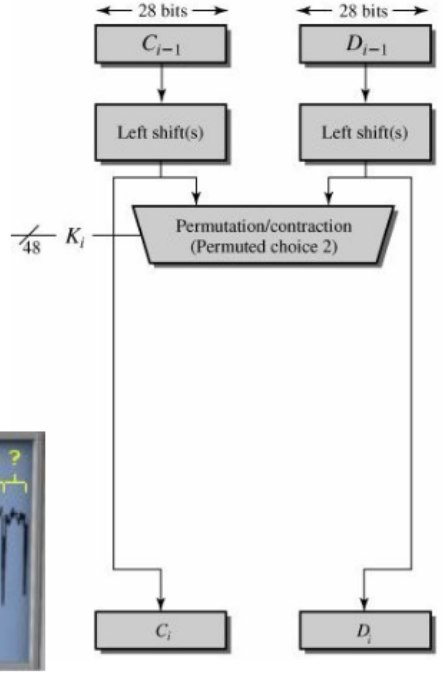


- SPA DES trace showing differences in power consumption of different microprocessor instructions

SPA on DES

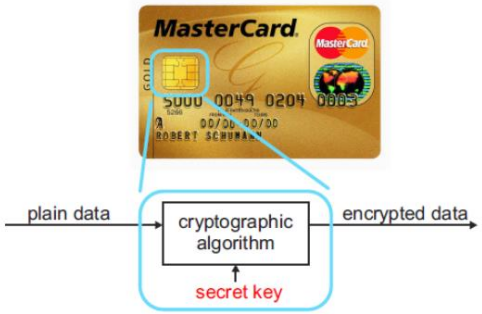
- SPA can reveal sequence of instructions executed
- It can be use to break cryptographic implementations in which the execution path depend on the data being processed
  - DES key schedule
  - DES permutations
  - Comparisons
  - Multipliers
  - Exponentiators

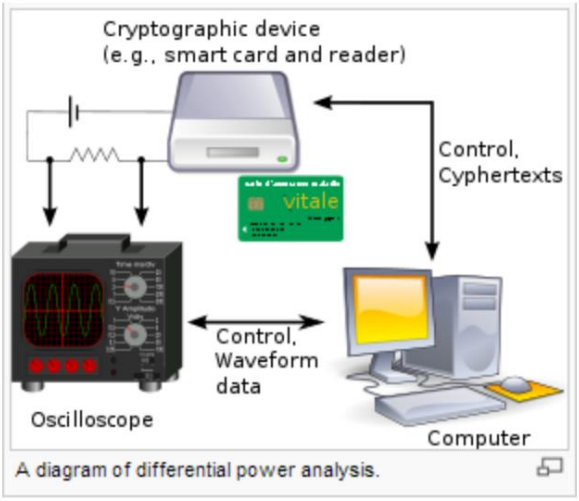




Differential Power Analysis

- SPA targets variable instruction flow
- DPA targets data-dependence
  - Different operands present different power
- Difference between smart cards and FPGAs
  - In smart cards, one operation running at a time
    - → Simple power tracing is possible
  - In FPGAs, typically parallel computations prevent visual SPA inspection → DPA

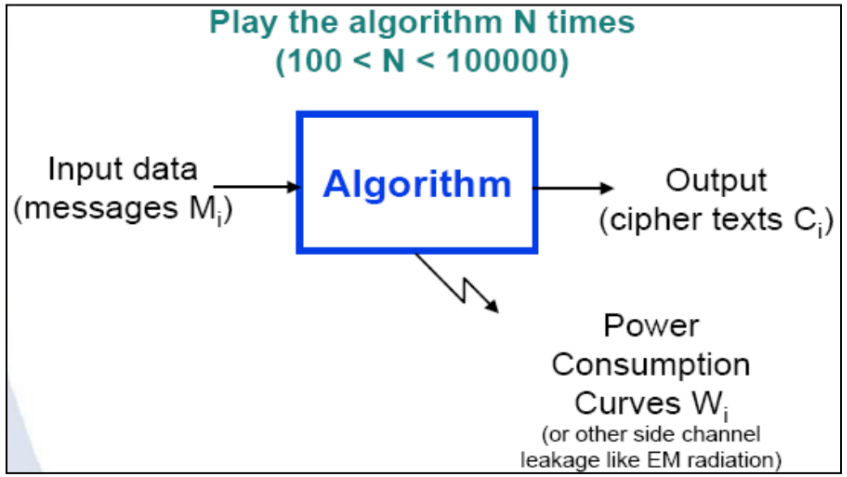




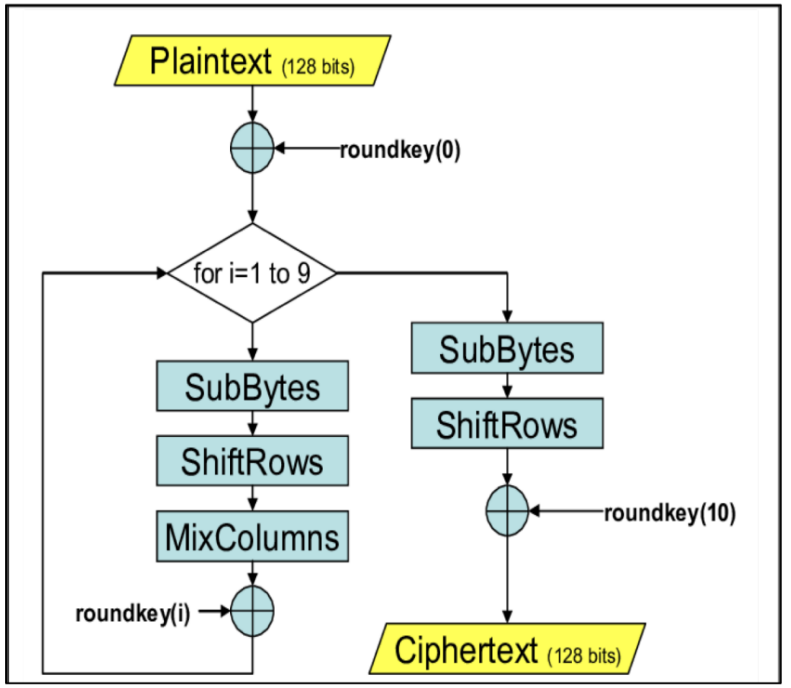
A diagram of differential power analysis.

DPA

- DPA can be performed on any algorithm that has the operation $\beta=S(\alpha \oplus K)$,
  - α is known and K is the segment key



The waveforms are captured by a scope and sent to a computer for analysis

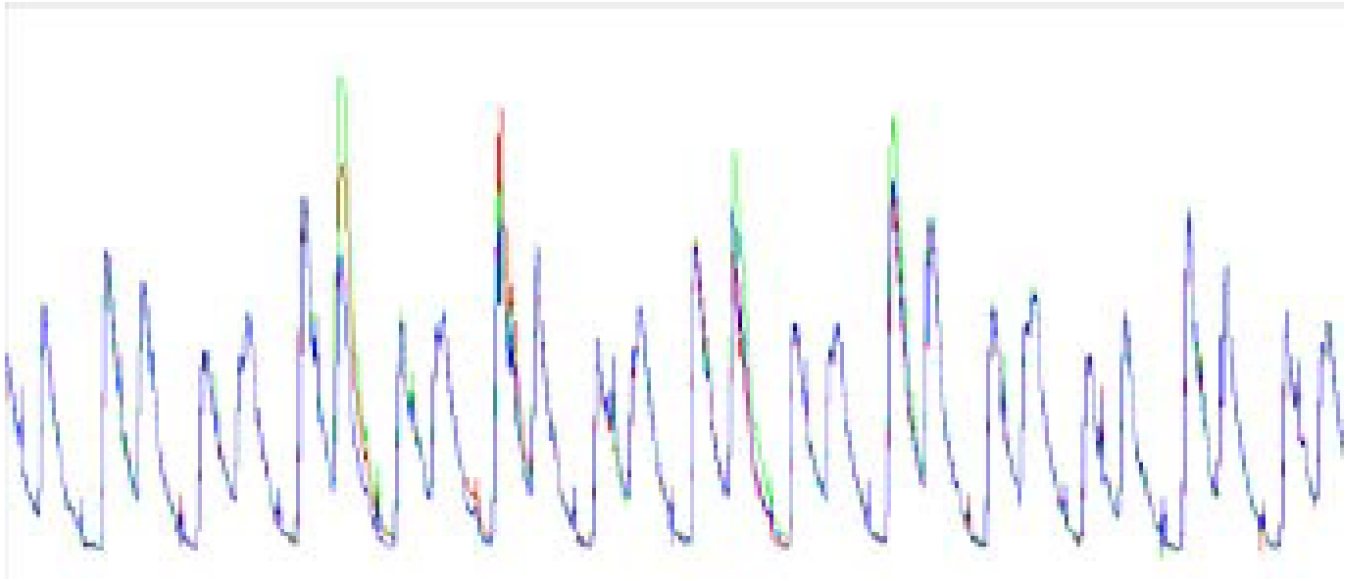


Assumption: Either Plaintext or Cipher is known

DPA

- After data collection, what is available ?
  - N plain and/or cipher random texts

    00 B688EE57BB63E03E
    01 185D04D77509F36F
    02 C031A0392DC881E6 …
  - N corresponding power consumption waveforms



DPA

- Assume the data are processed by a known deterministic function f (transfer, permutation…)
- Knowing the data, one can re-compute off line its image through f

- Now select a single bit among M' bits (in M' buffer)
- One can predict the true story of its variations

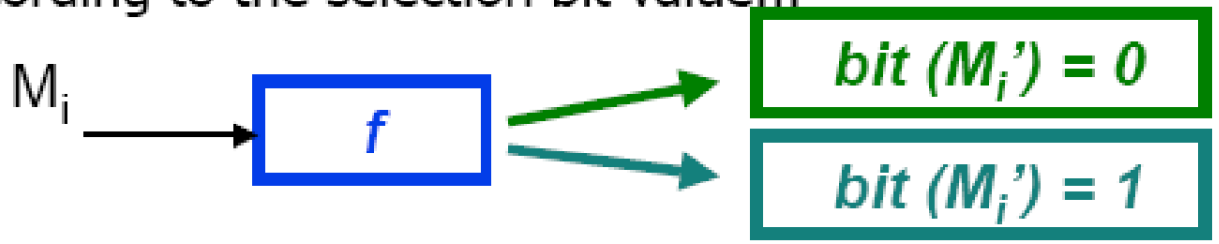
i	Message	bit	
0	B688EE57BB63E03E	1	
1	185D04D77509F36F	0	
2	C031A0392DC881E6	1	….

The bit will classify the wave w_i

- Hypothesis 1: bit is zero
- Hypothesis 2: bit is one
- A differential trace will be calculated for each bit!

DPA

- Partition the data and related curves into two packs, according to the selection bit value…



0	B688EE57BB63E03E	1
1	185D04D77509F36F	0
2	C031A0392DC881E6	1

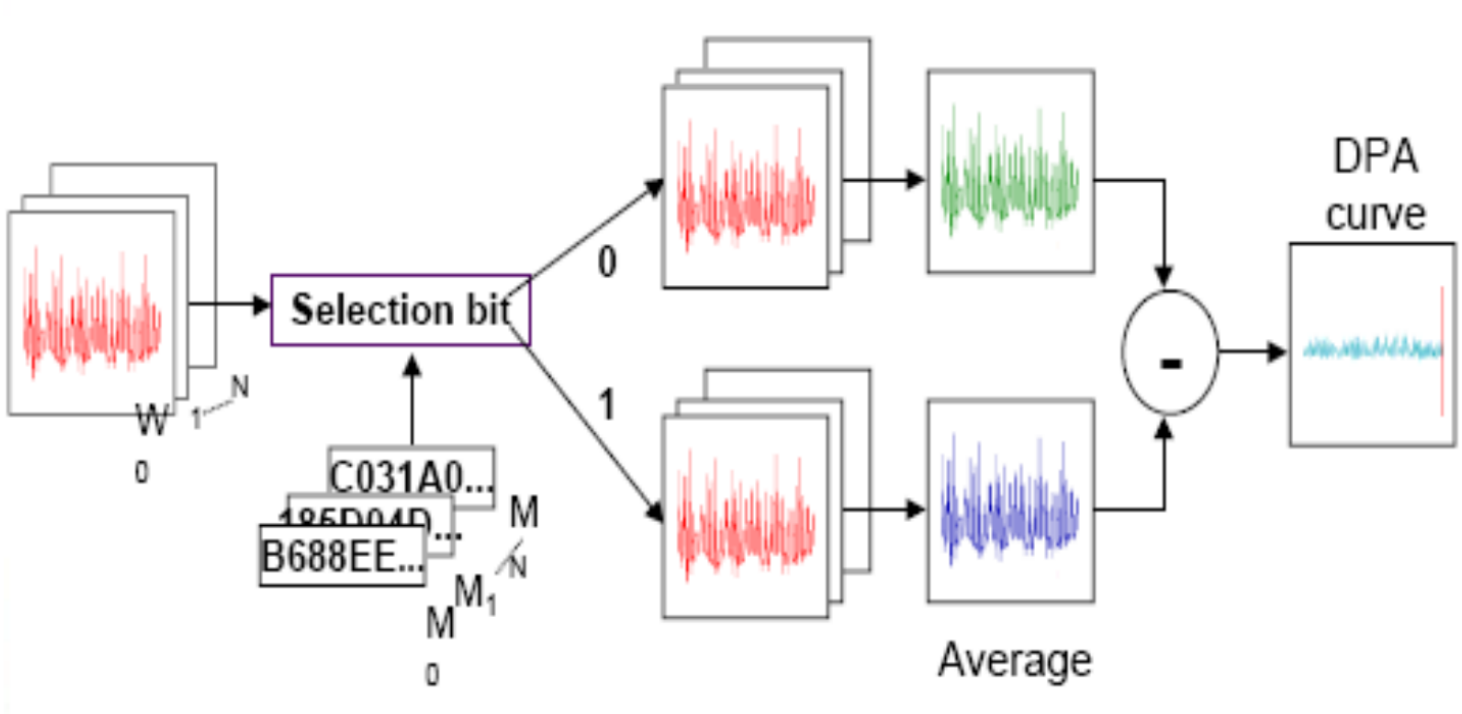
…

- Sum the signed consumption curves and normalise
- <=> Difference of averages

  ($N_0 + N_1 = N$)

$$DPA = \frac{\sum W_1}{N_1} - \frac{\sum W_0}{N_0}$$

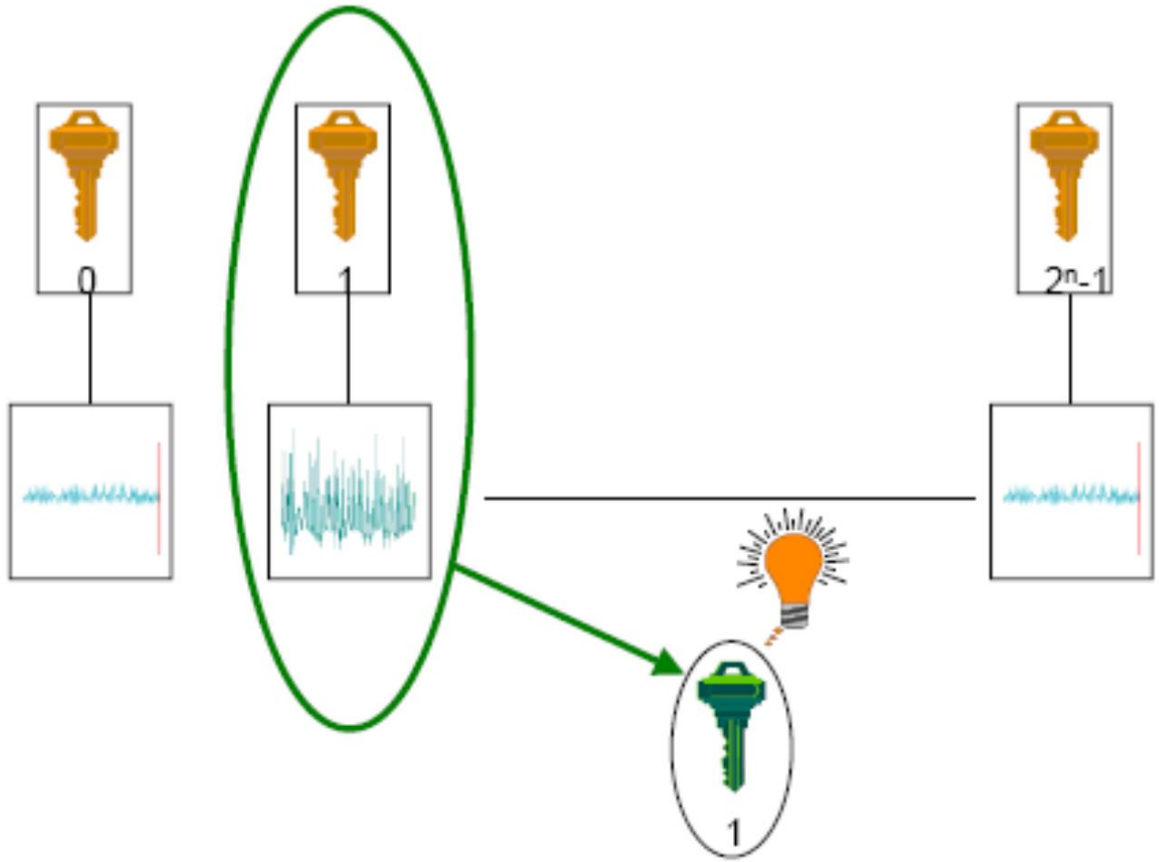
DPA



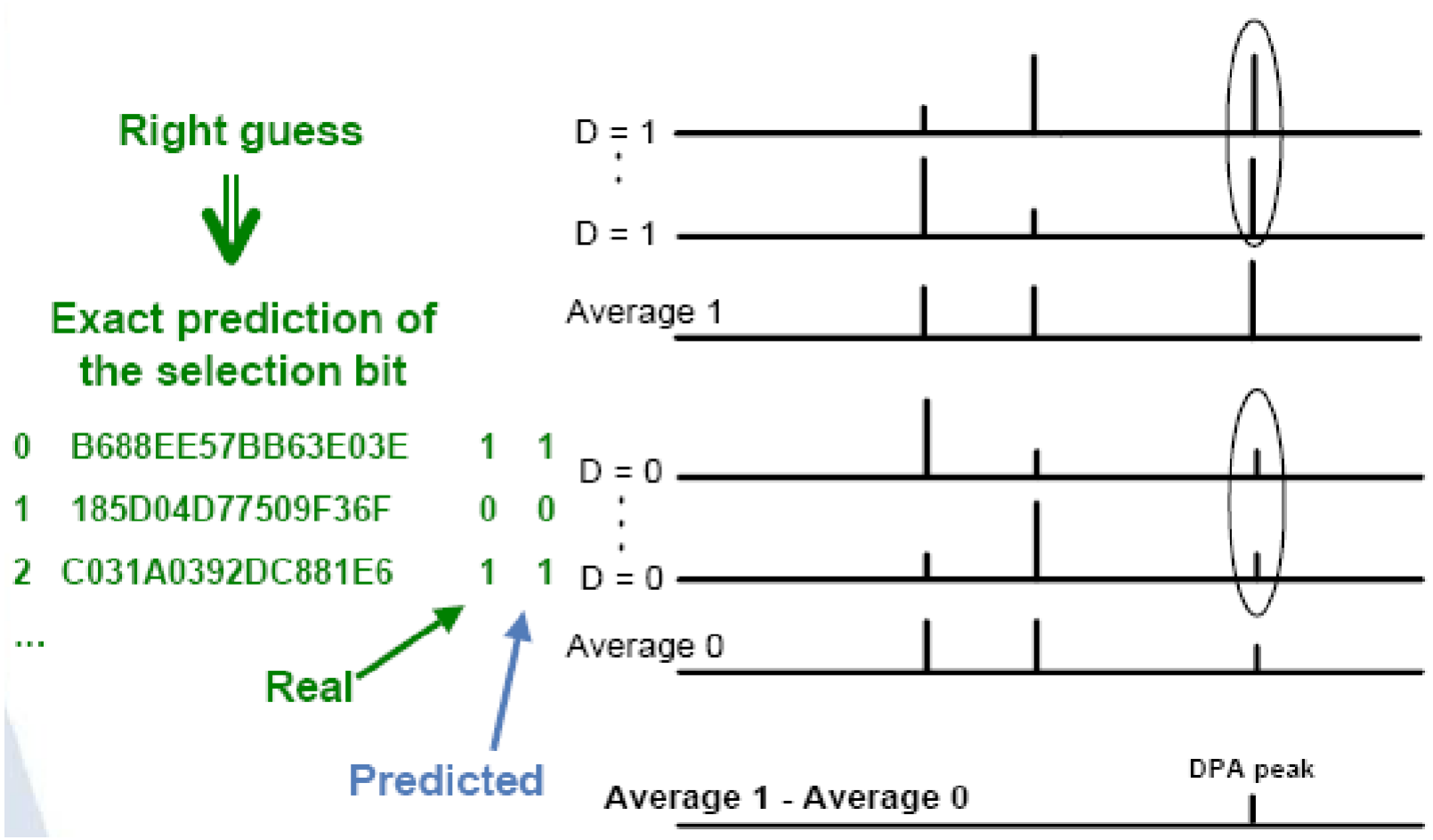
$$\Delta_n = \frac{\sum_{w_i \in S_0} w_i}{|S_0|} - \frac{\sum_{w_i \in S_1} w_i}{|S_1|}$$

DPA

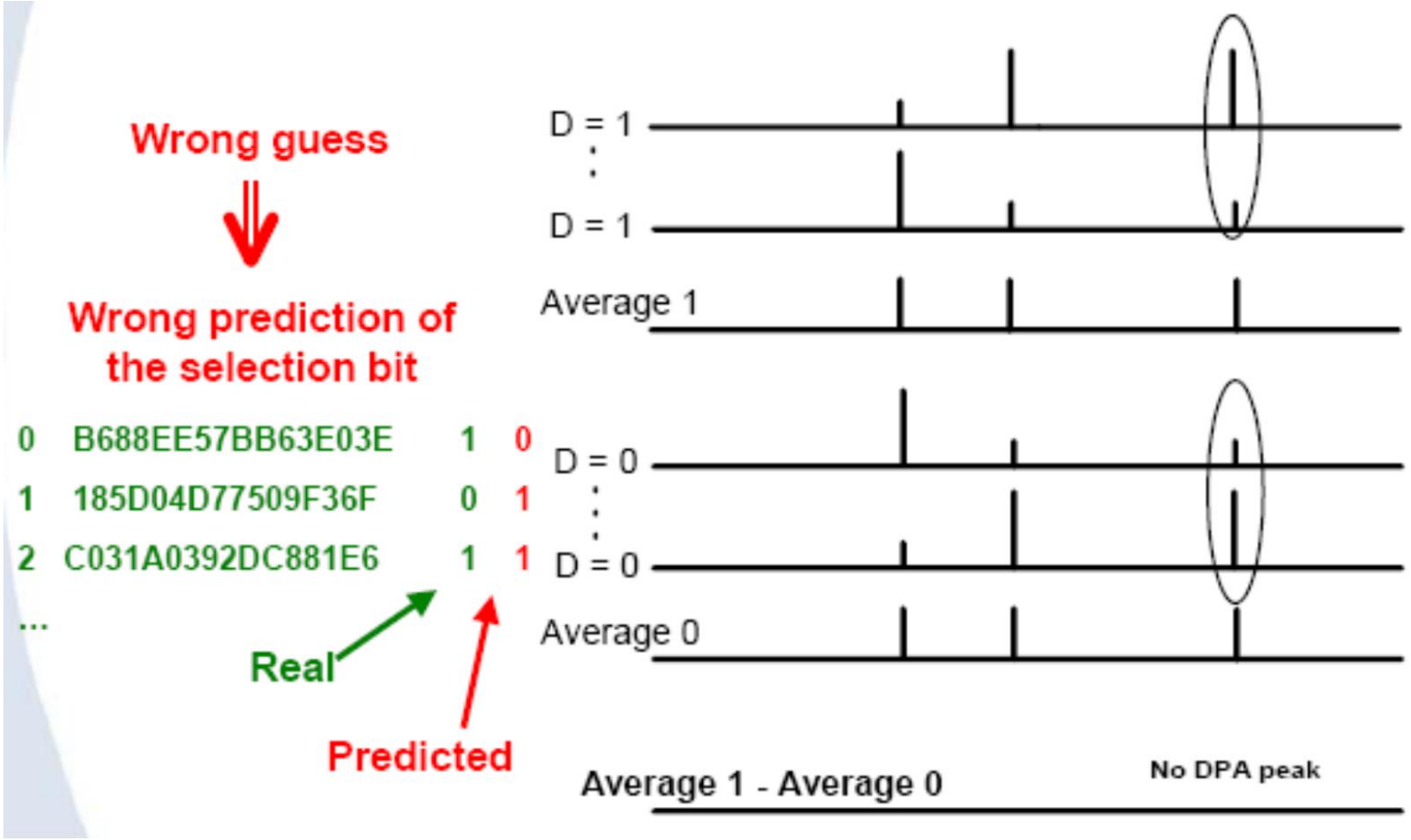
- The right guess provides the highest spikes !



DPA


Right guess
Exact prediction of the selection bit
0 B688EE57BB63E03E 1 1
1 185D04D77509F36F 0 0
2 C031A0392DC881E6 1 1
...
Real
Predicted
D = 1
D = 1
Average 1
D = 0
D = 0
Average 0
Average 1 - Average 0
DPA peak

DPA


Wrong guess
Wrong prediction of the selection bit
0 B688EE57BB63E03E 1 0
1 185D04D77509F36F 0 1
2 C031A0392DC881E6 1 1
...
Real
Predicted
D = 1
D = 1
Average 1
D = 0
D = 0
Average 0
Average 1 - Average 0
No DPA peak

Typical Power Traces

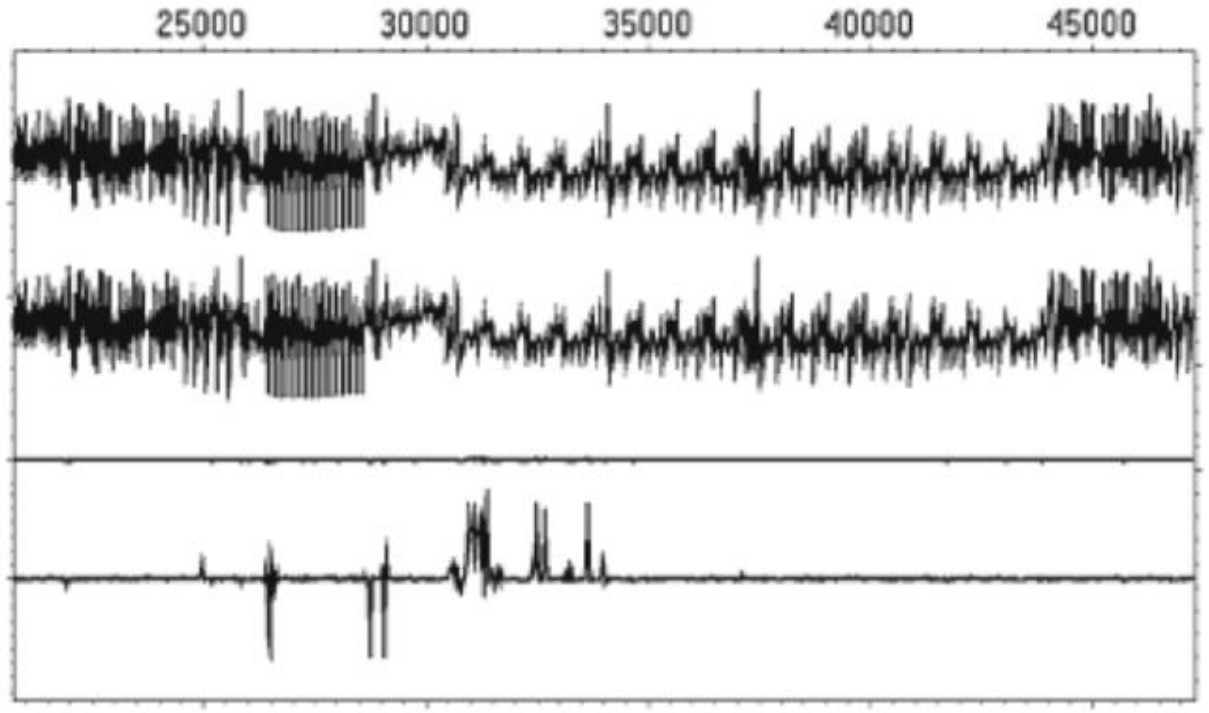


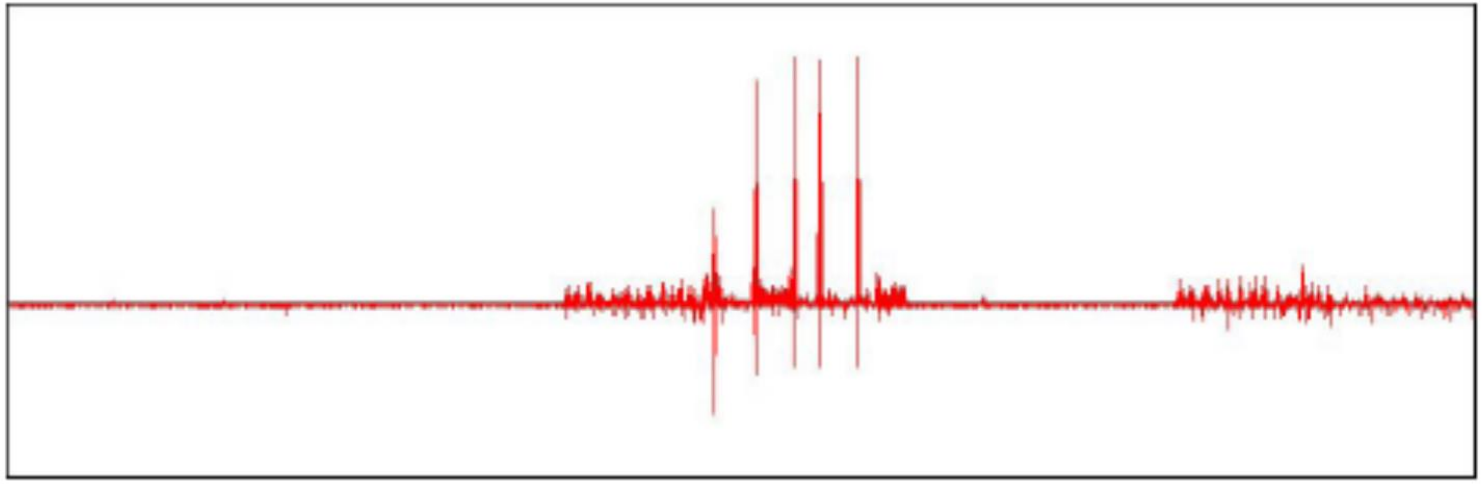
Figure 4: Typical DPA result. This example shows correlation.

(From: Intro to Differential Power Analysis[1])

- Typical DPA result showing the average of two sets of traces on the first two lines.
- The difference of these two sets is shown on the third line.
- The fourth line shows the same trace magnified by a factor of 15.
- This shows that there is statistical correlation between the two sets. If there was no correlation, the difference would be zero, or close to zero.

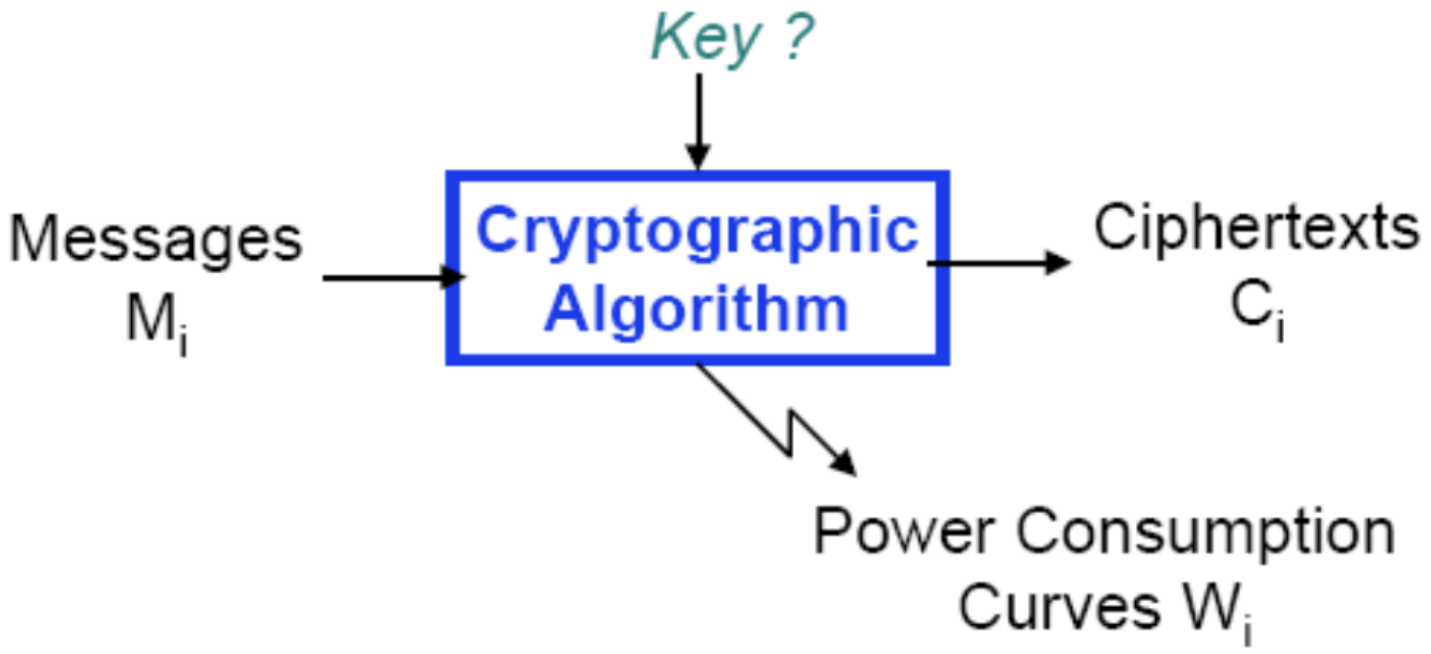
DPA

- The DPA waveform with the highest peak will validate the hypothesis



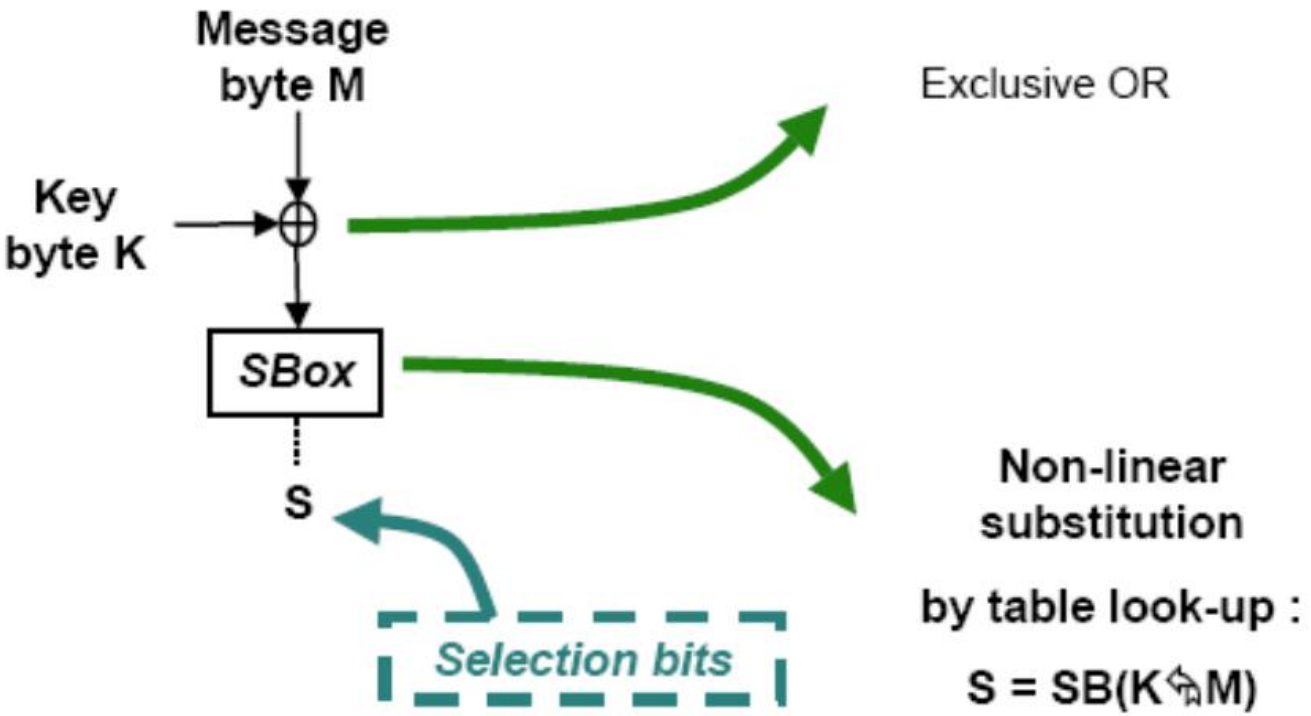
Attacking a secret key algorithm

- DPA works thanks to the perfect prediction of the selection bit
- How to break a key ?



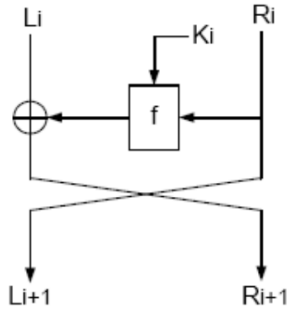
Typical DPA Target

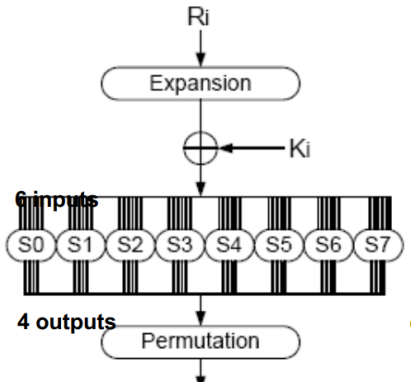
- Basic mechanism in Secret Key algorithms (AES, DES…)



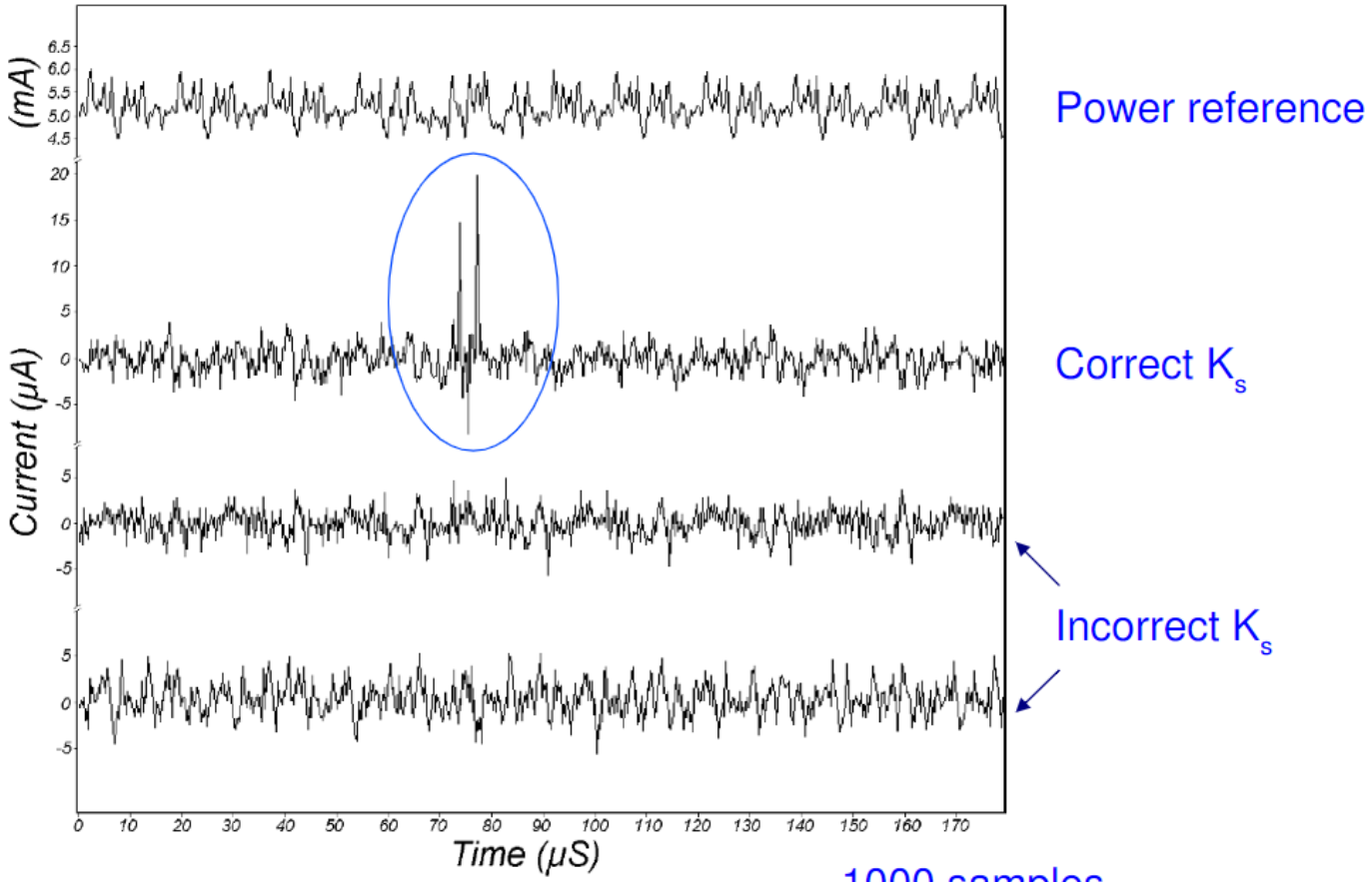
DPA on DES

- Assumption: Attacker presumes detailed knowledge of the DES
- Divide-and-conquer strategy, comparing powers for different inputs
  - Record large number of inputs and record the corresponding power consumption
  - Start with round 15 -- We have access to R_{15}, that entered the last round operation, since it is equal to L_{16}
  - Take this output bit (called M'_i) at the last round and classify the curves based on the bit
    - 6 specific bits of R_{15} will be XOR'd with 6 bits of the key, before entering the S-box
    - By guessing the 6-bit key value, we can predict the bit b, or an arbitrary output bit of an arbitrary S-box output
  - Thus, with 16 partitions, one for each possible key, we can break the cipher



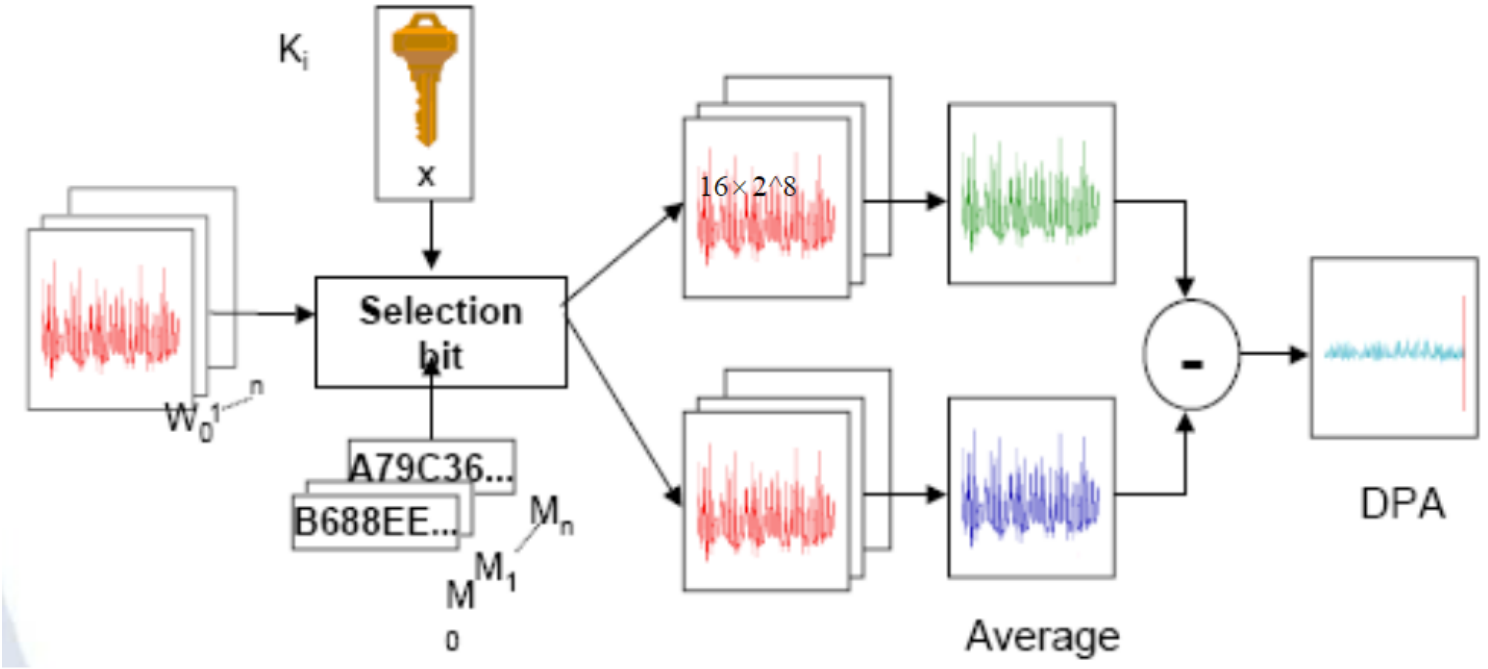


DPA Traces for DES



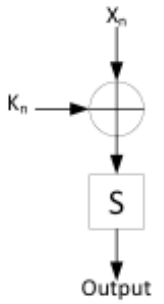
DPA on AES

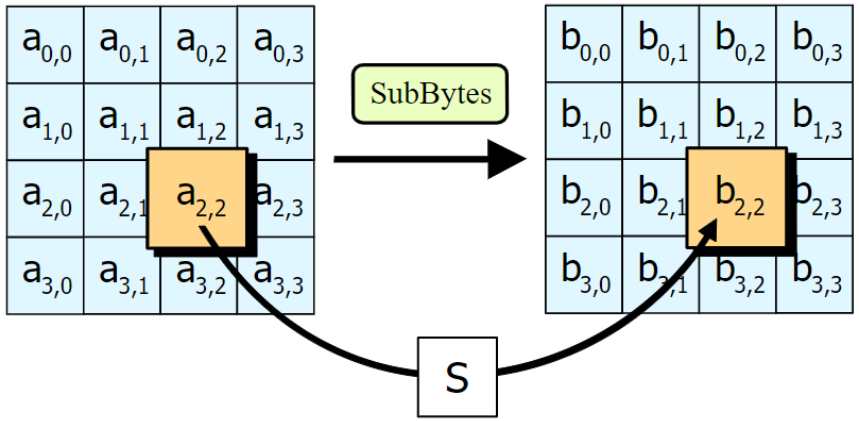
- Example : AES 128 bits key = 16 bytes K_i (i = 1 to 16)
  - Test 256 guesses per K_i with 256 DPA
  - 128 key bits disclosed with 16 x 256 = 4096 DPA (<< 2^{128} !)



Attacks on S-Box

$$\text{Output} = S[X_n \oplus K_n]$$





- Where *S* is a look-up table and $\oplus$ is the XOR of a known input X_n and the encryption key K_n. To determine the value for K_n, we make several guesses for the value of K_n.

- The first set of traces falls into the set where the LSB of the output is '0'; the second set of traces falls into the set where the LSB of the output is '1'. The difference of the average of the two sets is then examined.

DPA Results

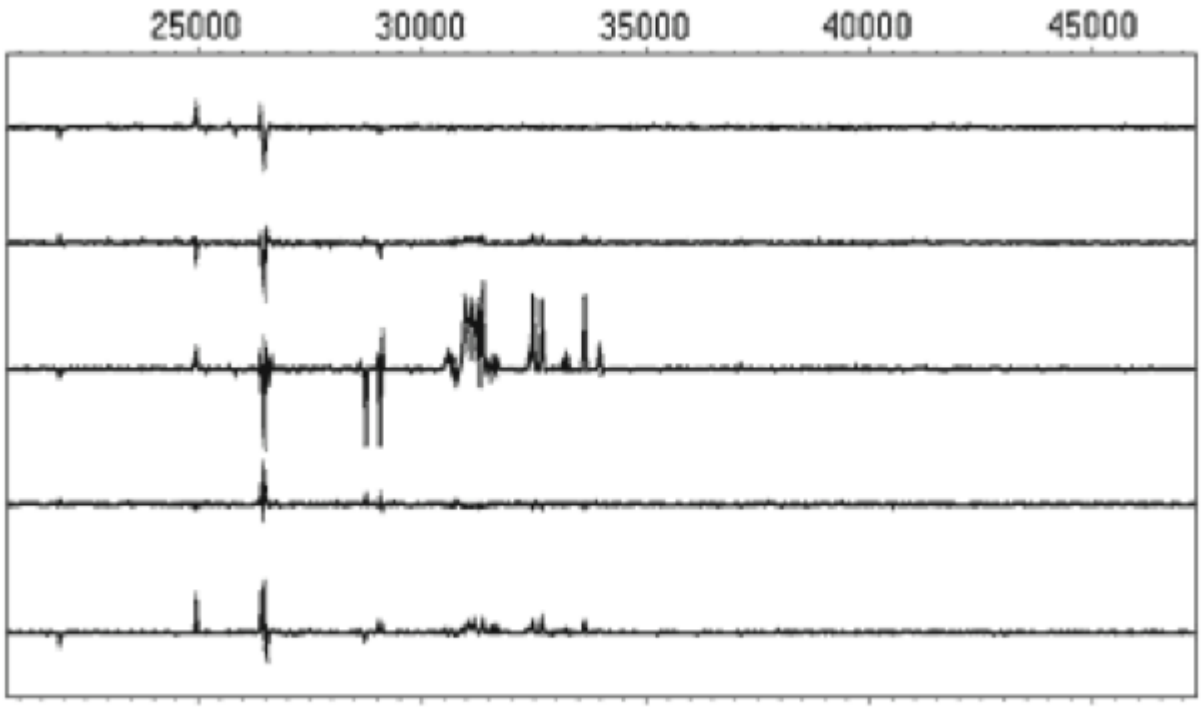
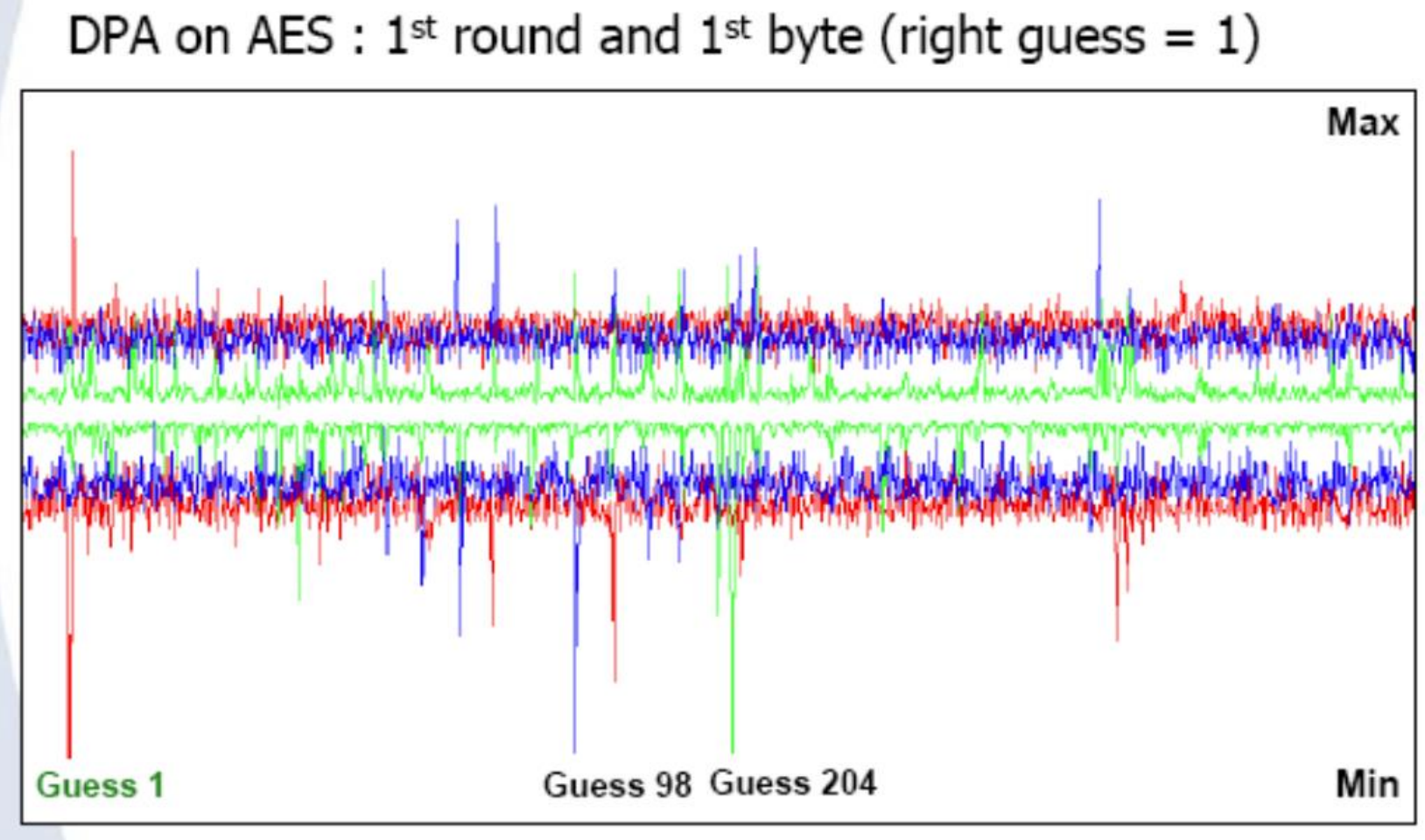


Figure 5: DPA result for different key values.

(From: Intro to Differential Power Analysis[1])

- The difference of the average of the two sets is then examined. Here, we have a trace showing the results of five different K_n values, where the correct key corresponds to the third trace.

Hypothesis Testing



Countermeasures

- Hiding -- reduce the SNR by either increasing the noise or reducing the signal
  - Noise Generators, 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
- Design Partitioning -- separate regions of the chip that operate on plaintext from regions that operate on ciphertext
- Physical Security and Anti-Tamper -- denial of proximity, access, and possession

Anti-DPA Countermeasures

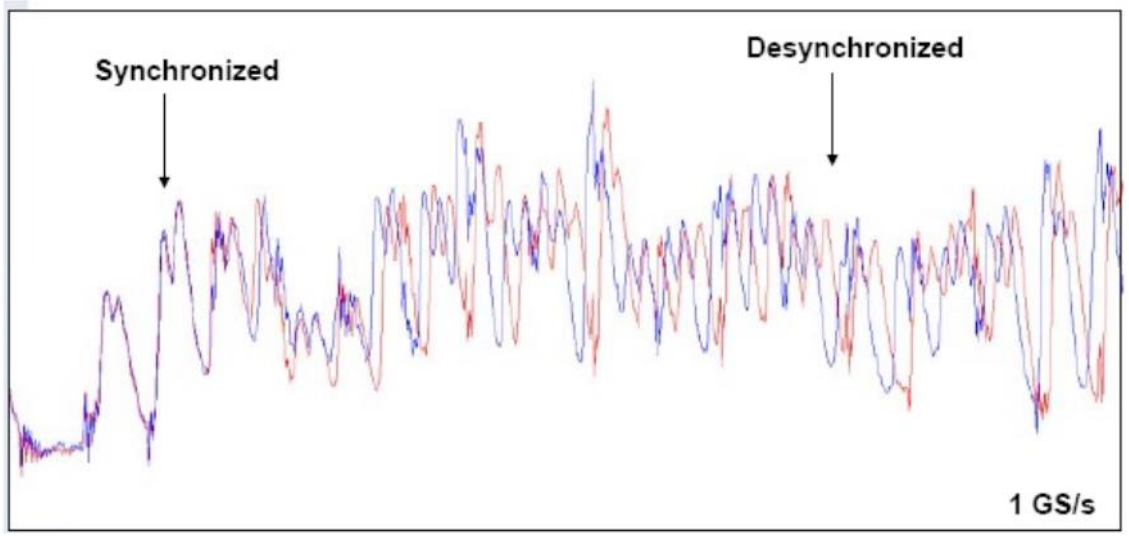
- Applicative counter-measures : make message free randomization impossible !
  - Fix some message bytes
  - Constrain the variable bytes (ex : transaction counter)
- Decorrelate power curves from data
  - by hardware : current scramblers (additive noise)
  - by software : data whitening
- Desynchronise the N traces (curves misalignment)
  - software random delays
  - software random orders (ex : SBoxes in random order)
  - hardware wait states (dummy cycles randomly added by the CPU)
  - hardware unstable internal clock (phase shift)

- DPA is powerful, generic (to many algorithms) and robust (to model errors)...
- ... but there are counter-measures !

GEMPLUS

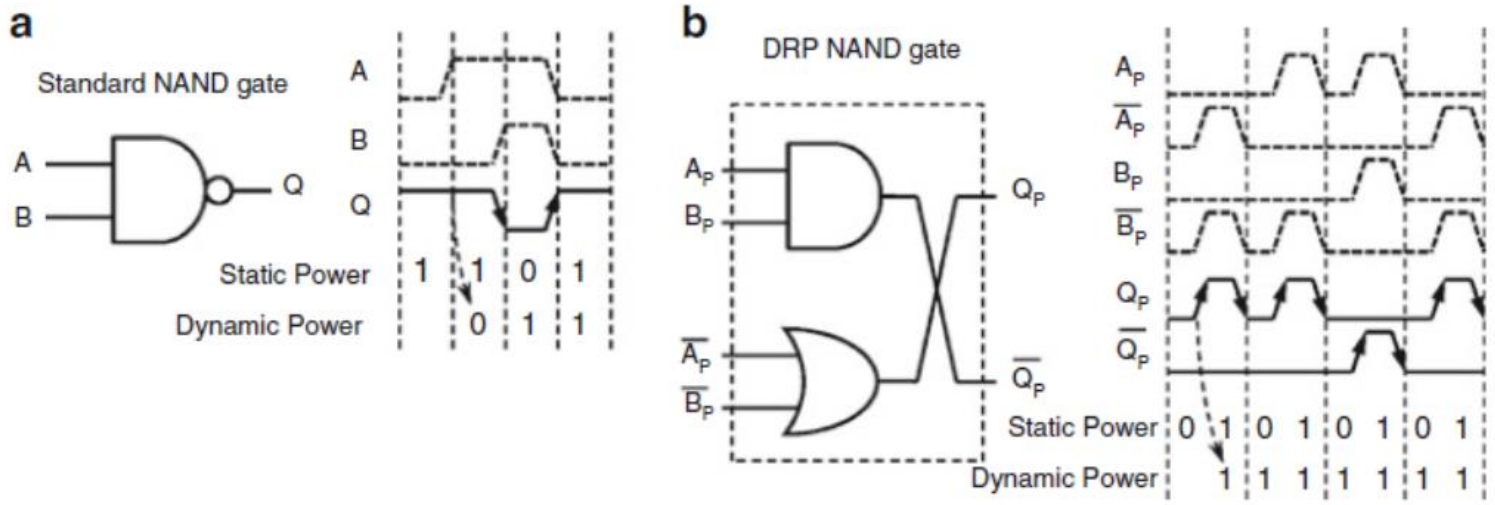
Anti-DPA

- Internal clock phase shift



Introduce noise to power consumption measurements

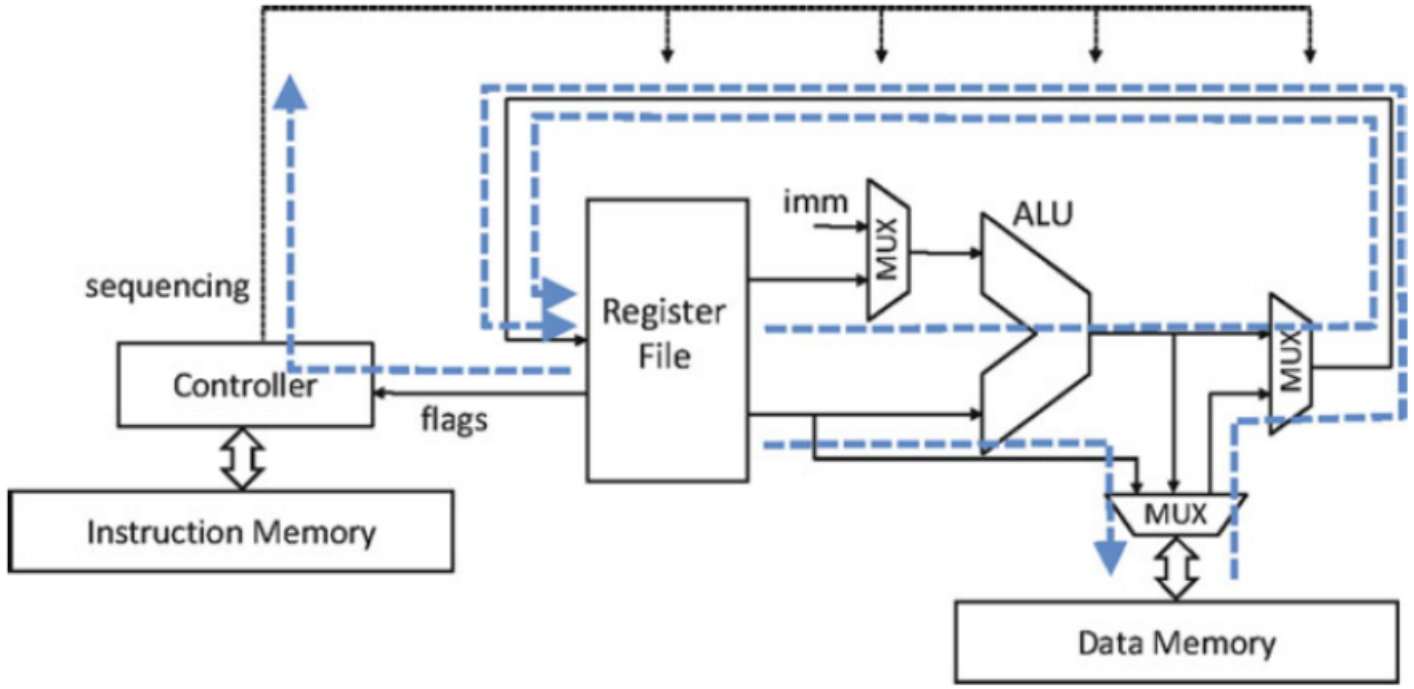
Dual Rail Precharge



- (a) A CMOS standard NAND has **data-dependent** power dissipation;
- (b) A DRP NAND gate has a **data-independent** power dissipation

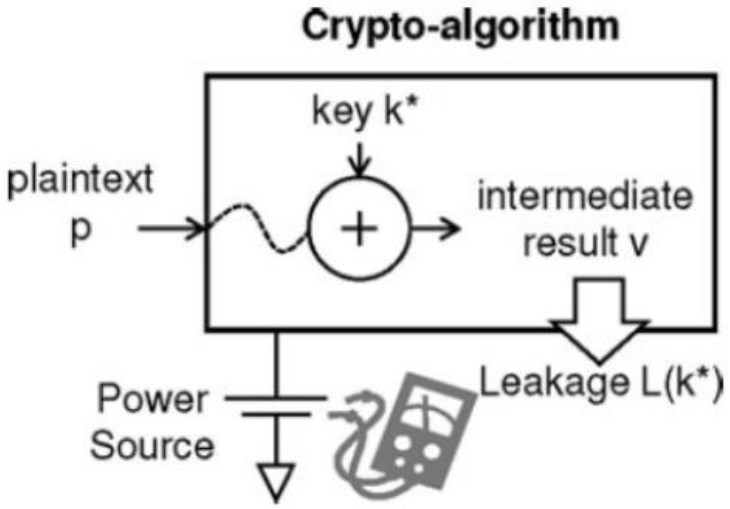
- DRP requires the execution of the direct and complementary data paths in parallel.

Side-Channel Leakage in Microcontroller



- Memory-store instructions
- Memory-load instructions
- Arithmetic instructions
- Control-flow instructions

Side-Channel Attacks on uController



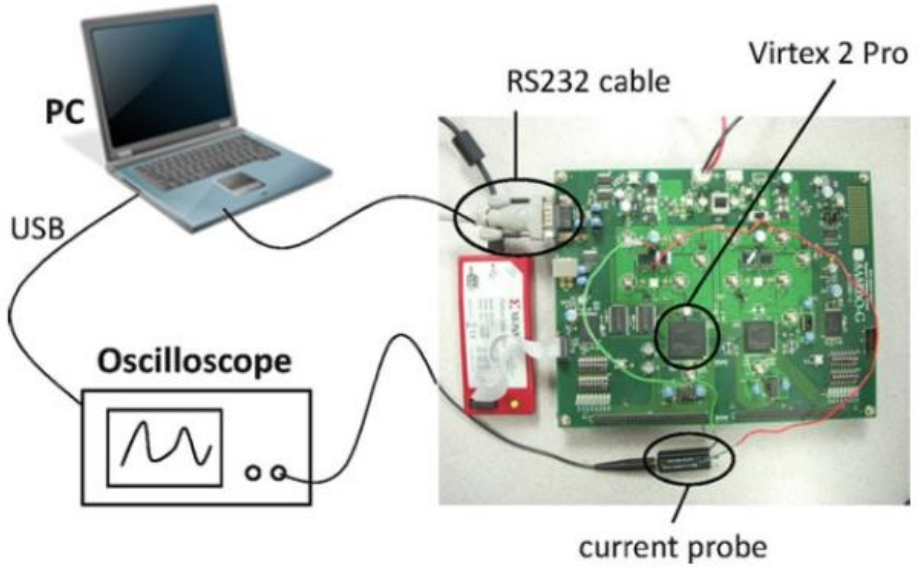
Objective: retrieve the internal secret key k^* of a crypto-algorithm

- The leakage caused by v is a function of the key value k^*, and it can be expressed as follows:

$$L(k^*) = f_{k^*}(p) + \varepsilon$$

The function f_{k^*} is dependent on the crypto-algorithm as well as on the nature of the implementation in hardware and software. The error ε is an independent noise variable.

Side-Channel Attacks on uController



- The PC sends a sample plaintext to the PowerPC on the FPGA for encryption. During the encryption, the digital oscilloscope captures the power consumption from the board. After the encryption is completed, the PC downloads the resulting power trace from the oscilloscope, and proceeds with the next sample plaintext.

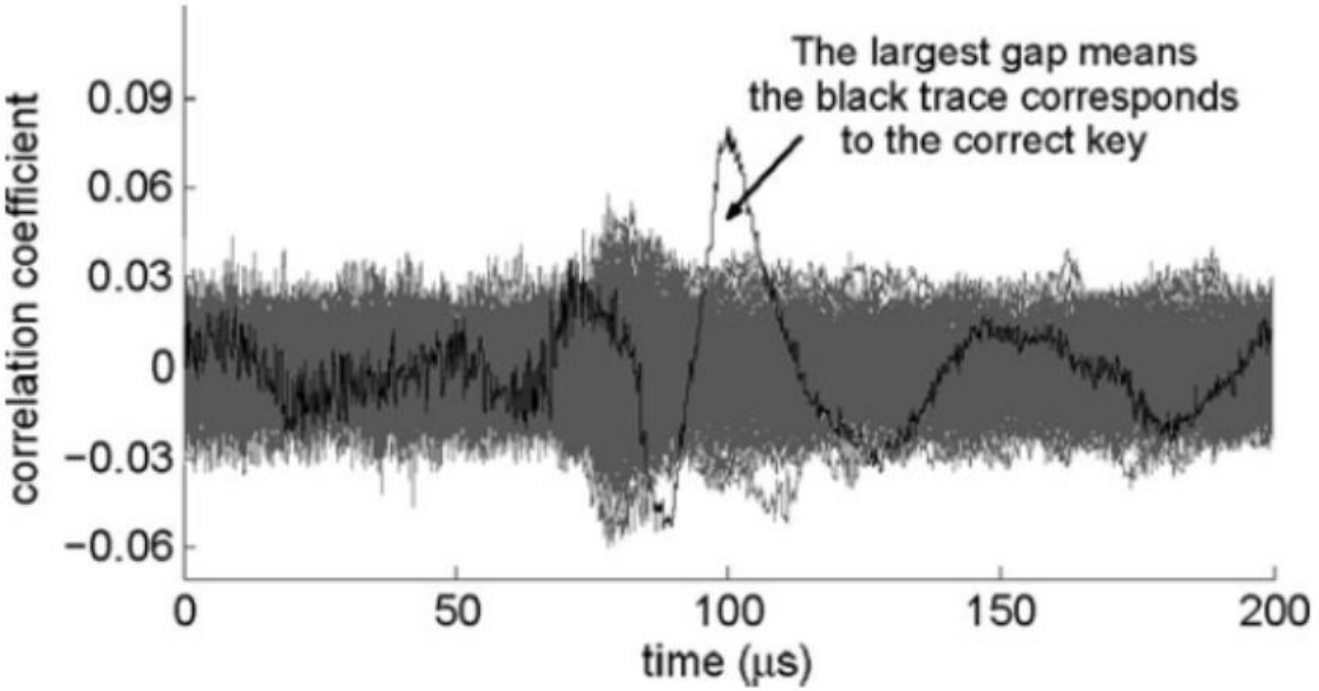
CPA: Correlation Power Analysis

- Two important aspects of a practical CPA:
  - The selection of the power model

    The power model is chosen so that it has a dependency on a part of the secret key. A good candidate is the output of the substitution step.
  - The definition of the attack success metric

    Measurements to Disclosure (MTD): the more measurements that are required to successfully attack a cryptographic design with side-channel analysis, the more secure that design is.

Practical Hypothesis Tests



- An example of 256 correlation coefficient traces. Around time 100 us, the black trace which corresponds to the correct key byte emerges from all the other 255 traces.

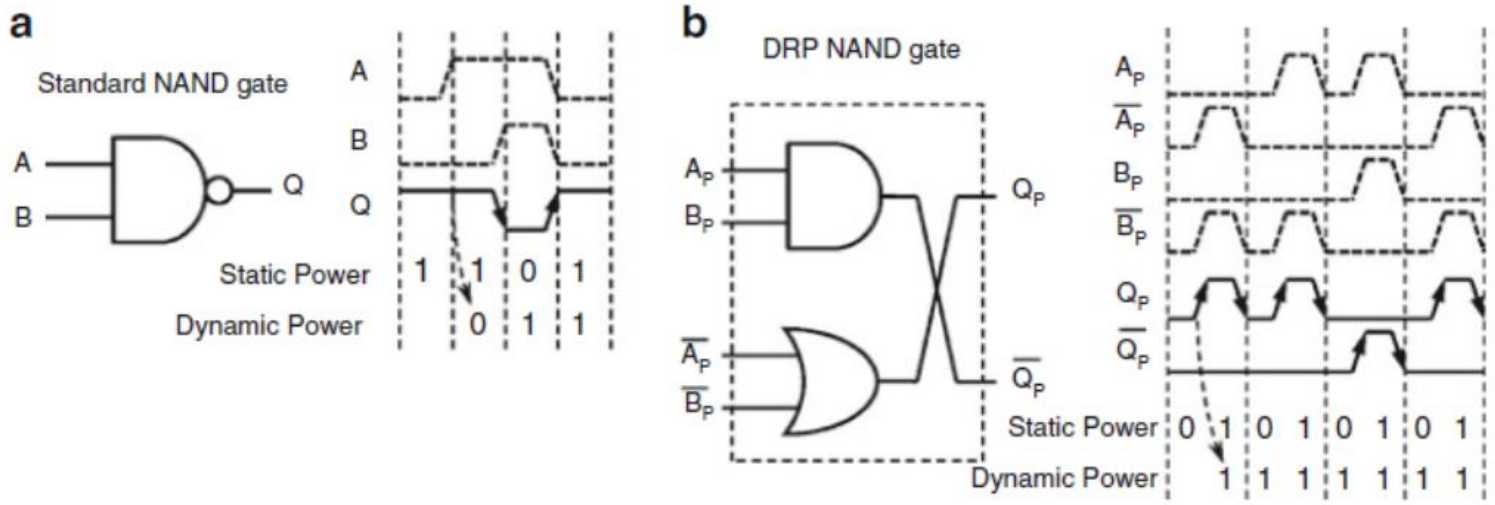
Countermeasures

- Two different kinds of countermeasures:
  - Algorithm-Level Countermeasures

    Transform the C program so that the generation of dangerous side-channel leakage is avoided.
  - Architecture-Level Countermeasures

    Create a better microcontroller, for example using special circuit techniques, so that no side-channel leakage is generated.

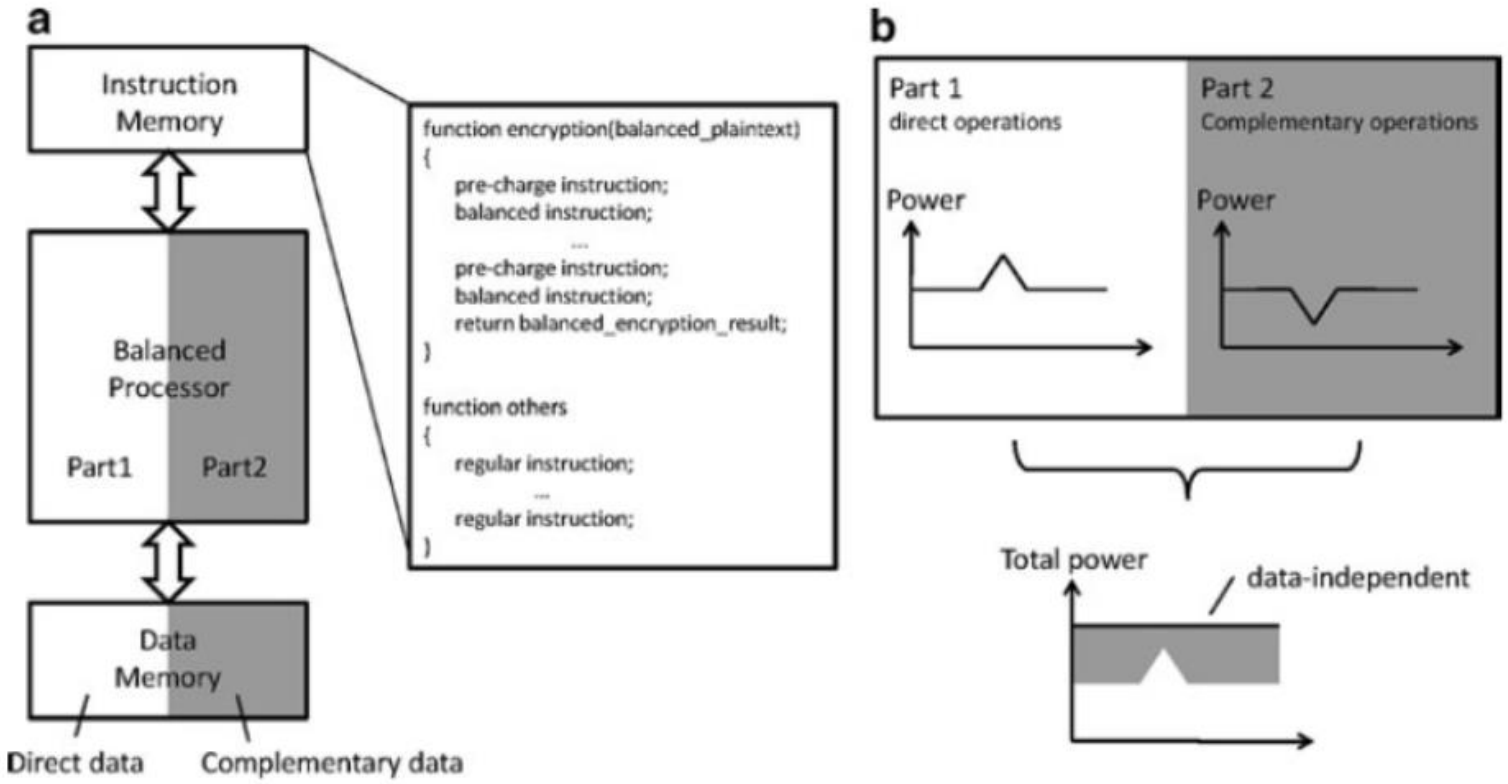
Dual Rail Precharge



- (a) A CMOS standard NAND has **data-dependent** power dissipation;
- (b) A DRP NAND gate has a **data-independent** power dissipation

- DRP requires the execution of the direct and complementary data paths in parallel.

Porting DRP into Software



- (a) Concept of balanced processor and VSC programming;
  (b) The balanced processor does not show side-channel leakage

A good youtube resource

- https://www.youtube.com/watch?v=OlX-p4AGhWs&t=3638s