(Side channel Analysis, and Counterfeiting)

Side-channel Attacks

- **Side channel attacks (SCA):**
  - Monitor/measure chip’s physical characteristics (power, current, timing, EM radiation, etc.) during its normal operation
  - Perform data analysis to learn information

- Side-channel attacks exploit the leakage of secret information through a physical modality when an application is being executed on a system.

- Features of side channel attacks
  - SCA is non-invasive and passive
  - SCA combined with other “active” methods
    - Control the normal operation via (rare) input
    - Force abnormal operation (e.g. fault injection)

- Side-channel attacks are powerful and have been able to break most existing important cryptographic algorithms.

Side-channel Attacks: Sources of Side Channel

- **Measurable physical features:**
  - Power consumption or current
  - Timing or delay
  - Electromagnetic radiation
  - Photonic emissions
  - Acoustic noise of the system
  - Output signals

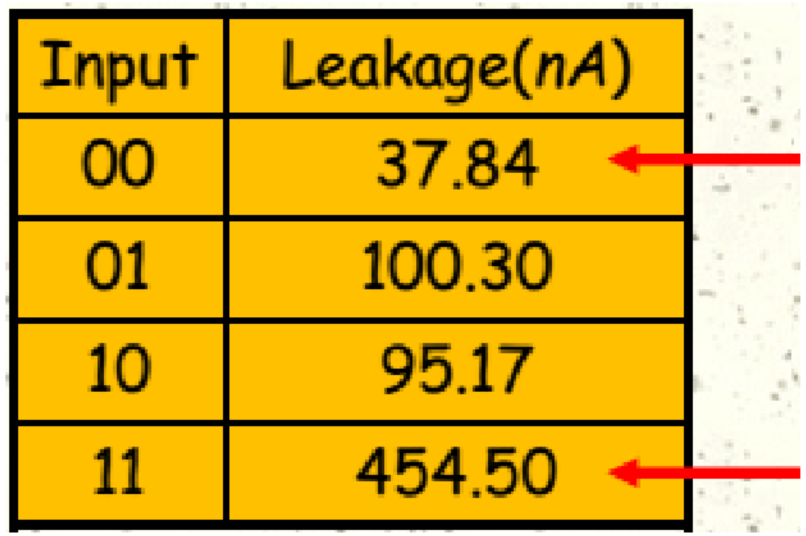
Side channel Attack: Power and Current Example

- Source of power consumption
  - Dynamic power- Needed for charge and discharge a capacitor
  - Leakage current- occurs even when system is ideal
  - Short circuit and others

- Why data may leak from power/current
  - Dynamic power: $P \propto C V^2 f$

    C: effective capacitance of switching activity

    (logical 0-> 1or Logial1->0)
  - Leakage current: depend on the input vectors

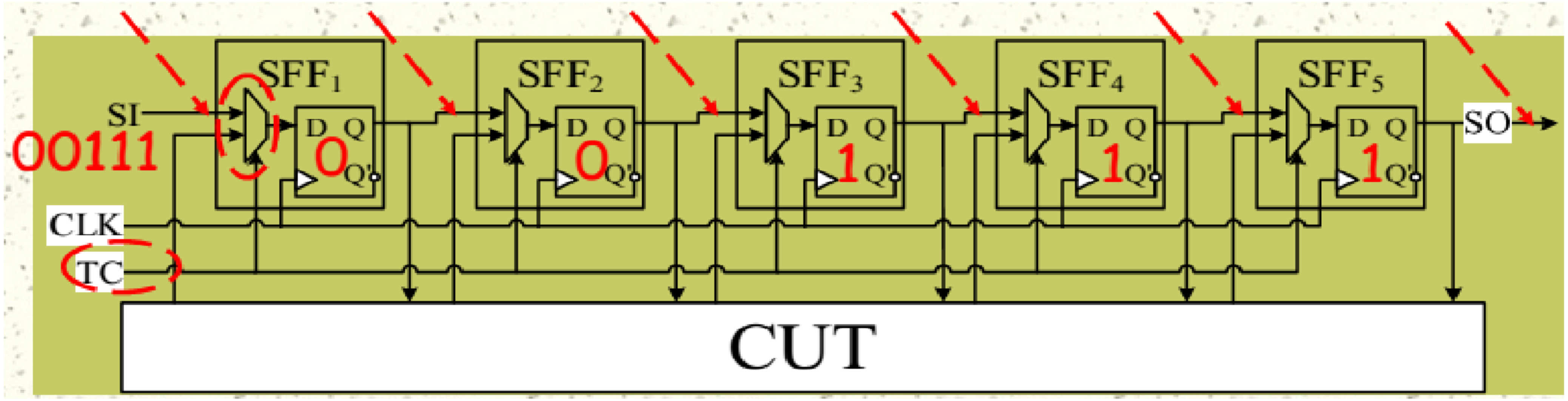


Input	Leakage(*nA*)
00	37.84
01	100.30
10	95.17
11	454.50

Leakage current in a 2-input NAND gate

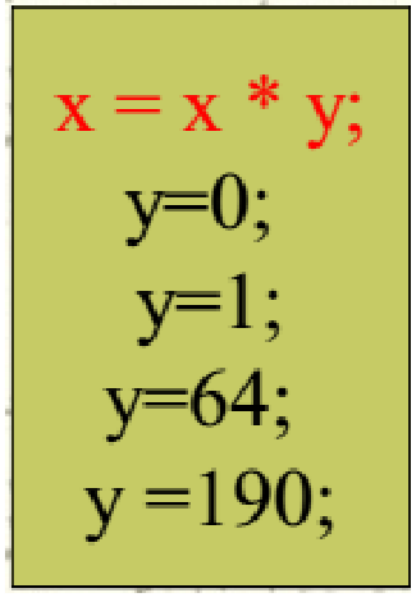
Side channel Attack: Scan Chain Example

- Source of scan chain channel
    - On-chip registers (scan flip flops)
- Why data may leak from scan chain
    - System internal state can be read directly from scan out port during testing mode
    - TC=0, provide primary input to the system
    - TC=0, run for one or more clock cycles
    - TC=1, test mode to capture internal state from scan out
    - Repeat to collect more information

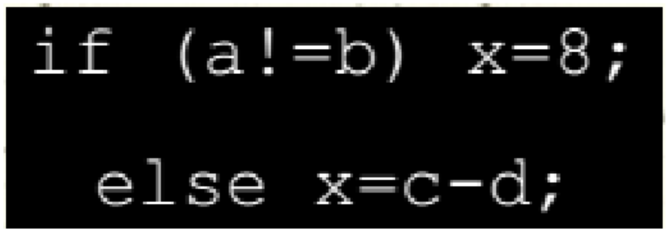


Side channel Attack: Timing or Delay Example

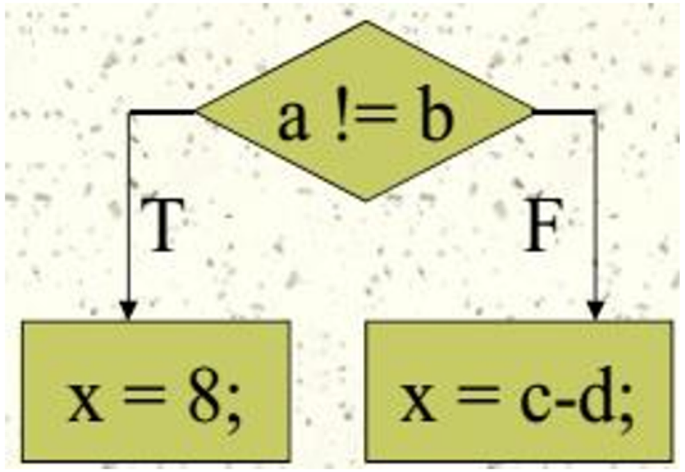
- Source of timing and delay
  - Execution time required to complete an operation.
- Why Data may leak from timing/delay
  - Execution time of different operations may be different.
  - Execution time of same operation for different operand and in different conditions may also be different
  - **Example:**
    - Control flow
    - Data dependency
    - Cache miss
    - Pipeline stall



```
x = x * y;
y=0;
y=1;
y=64;
y =190;
```



```
if (a!=b) x=8;
 else x=c-d;
```



Side channel Attack: RSA Example

- For example, In RSA encryption the value of m^e can be calculated in two ways:
  - By naive approach- it require (e-1) multiplications.
  - Square and multiply - It reduces the overhead and generally used.
- To calculate the m^{1026} by naive approach takes 1025 multiplication while square and multiply algorithm requires only 11 multiplication.

- **Motivational Example:**
  - How do we compute a^{16}?
    - $a^{16} = a*a*a*\ldots*a$: multiply 15 times
    - $a^{16}=(a^8)^2= (((a^2)^2)^2)^2$: Square four times

  - How about a^{23}?
    - $a^{23}=(a^{16})(a^4)(a^2)(a^1)$: Square four times : no need to do 22 multiplications

Side channel Attack: Example (Cont…)

- **Square and Multiply Algorithm:**

  - **Goal:** Compute a^e(mod n)

```
Convert e to binary:   $k_s k_{s-1} \ldots k_1 k_0$
b = 1;
for (i=s; i>=o; i--)
{
        b = b*b (mod n);
        if ($k_i$== 1)
            b = b * a (mod n)
}
return b;
```

Observable side channel information during hardware execution: current, power, timing, …

The value of bit k_i determines whether this non-trivial operation will be required.

Counter Measures against the Side Channel

1. **Leakage Reduction:** decrease the dependency between the side-channel traces (i.e. delay, power consumption etc.) and the secret information.

   Dependency between the timing information and the secret exponent can be reduced by performing ‘‘dummy’’ multiplication operations.

2. **Noise Injection:** The SNR of the side channel information can be reduced by injecting artificial noise, so It is more difficult for an attacker to retrieve the secret key from the noisy side channel.

3. **Key Update:** Key is updated before the amount of leaked information breaches a predefined level.

- ***All these existing approaches does not provide full protection from the SCA.***

Counterfeiting Attack

- It means making ICs by illegally copying or stealing an authentic blueprint/IC during one of the design, synthesis, or production phases.

- In recent years, because of technological advances in 3-D packaging, fake ICs are hard to distinguish from the real ones.

- Although the common incentive for selling fake ICs can be
  - Financial
  - Insertion of hardware Trojans in fake ICs to makes them a real security threat for the whole system

- The suppliers of the original components suffer financial and reputation loss because poor performance of fake products impacts the overall system performance/reliability.

Counter Measures against Counterfeiting

1. Hardware Metering and Auditing
2. IC Fingerprints or PUFs
3. Device Aging Models/Sensors
4. IP Watermarking

Counter Measures: Device Aging Models/Sensors

- IC lifetime is influenced by a variety of phenomena, such as negative temperature bias instability (NBTI), hot carrier injection, and electromagnetic migration.

- By employing sensors in ICs an estimate of chip lifetime can be found by measuring these phenomena.

- This estimate would prevent counterfeiters from selling used chips as new ones.

- This estimate also measure the previous usage of a device and detects its authenticity.

Thank You!