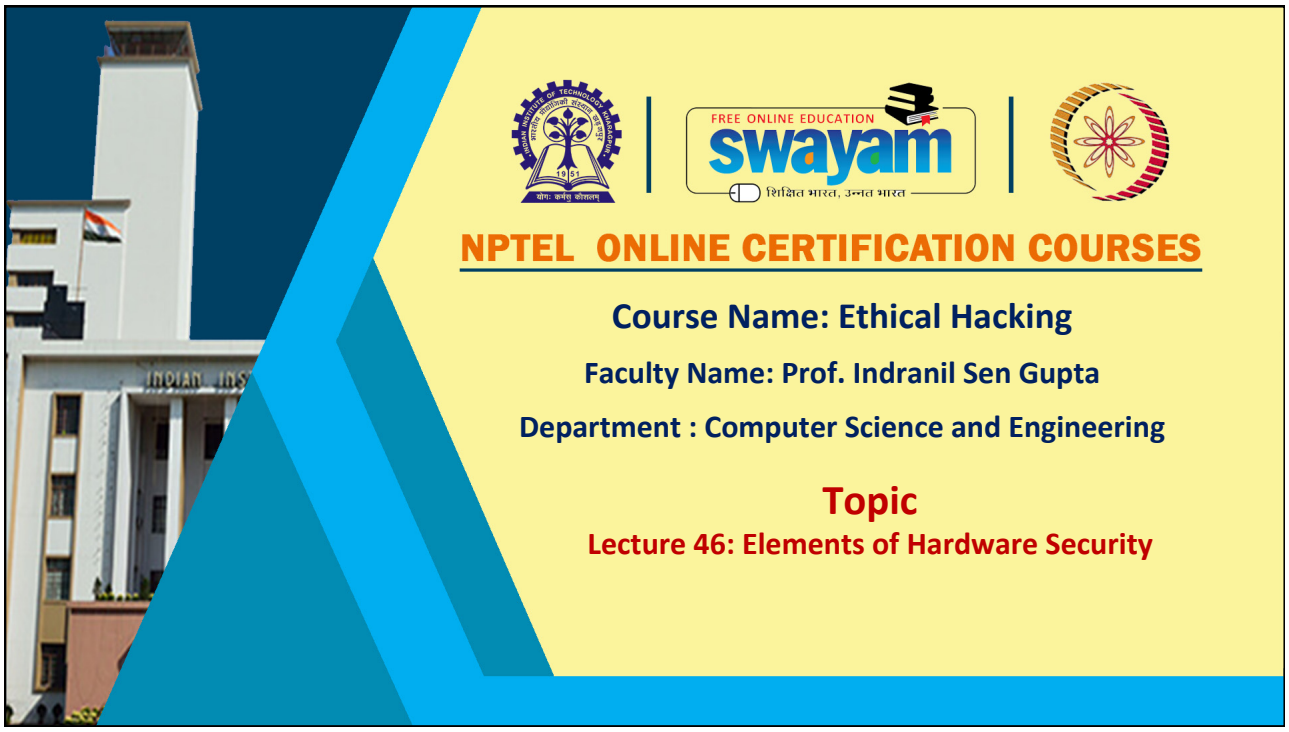


NPTEL ONLINE CERTIFICATION COURSES

Course Name: Ethical Hacking

Faculty Name: Prof. Indranil Sen Gupta

Department : Computer Science and Engineering

Topic

Lecture 46: Elements of Hardware Security

CONCEPTS COVERED

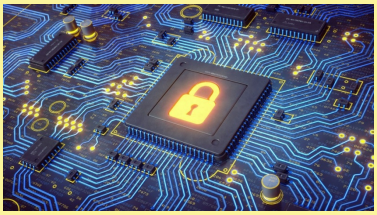
- Attacks on hardware
- Typical countermeasures

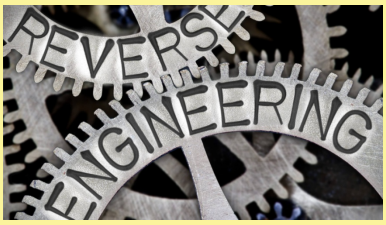
Introduction

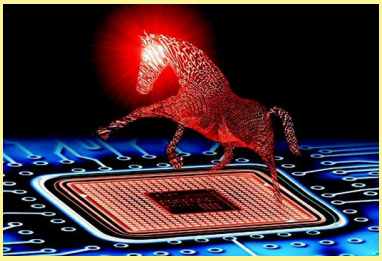
- How to we characterize hardware in the present context?
  - **Computer Hardware**, which includes processors, firmware and memory.
  - **Mobile Hardware**:
    - ❖ SIM Card
    - ❖ RFID
    - ❖ Smart Card
    - ❖ PUF

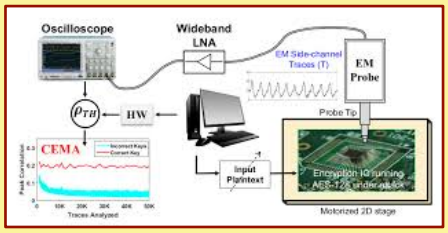
Attacks on Hardware

- **Physical Attacks**
  - Carried out on the actual device using hardware tools.
- **Planned Attacks**
  - Some vulnerability can be deliberately included in the hardware.
- **Stealing Secret Data**
  - Many hardware device carries confidential data.
  - International mobile subscriber identity and contact details in a SIM card.
  - Unique identification codes in RFID tags.
  - Secret key and other confidential information in a Smart Card.









Types of Attacks

a) Black Box Testing

- The attacker sends an input to the circuit and receives an output.
- Based on the input/output behavior, the attacker decides what kind of algorithm is used.
- This is an non-invasive type of attack.

b) Physical Probing

- The attacker plants a probe into the chip itself and reads data off the chip.
- This is an invasive attack, and requires sophisticated instrumentation.

Types of Attacks

- **Reverse Engineering**
  - The attacker acquires the device (say, smart card) and physically exposes the circuit.
  - Each layer of the circuit is removed and high resolution photographs are taken.
  - Invasive approach, and also requires very sophisticated instrumentation.
- **Side Channel Analysis**
  - The attacker measures sensitive parameters during normal operation of the circuit.
  - Based on the measurements, some secret values can be inferred.
  - This is a non-invasive kind of attack, and is the subject of intense research.


Typical Countermeasures to Prevent Hardware Attacks

a) Obfuscate data in registers and buses
   - Scramble, encrypt, etc.

b) Obfuscate the IC layout
   - Use 3D stacking, dummy circuitry, etc.

c) Add metal mesh on top of the circuit.
   - If the circuit is probed, it will cause a short and the stored data resets.

d) Countermeasures against side channel attacks.
   - Random noise generator, secret hiding, etc.

e) Physical unclonable function (PUF)
   - Can be used to design low-overhead security protocols.


Hardware Trojan

- Malicious logic inserted into a circuit without the knowledge of the designer / user.
  - Carries a trigger condition and a payload (may be malicious).
  - Very difficult to detect.
- Trojans can also be used for defensive purposes.
  - Any unauthorized change in the circuit will be detected.
  - Can be used for copyright protection (IC fingerprinting).

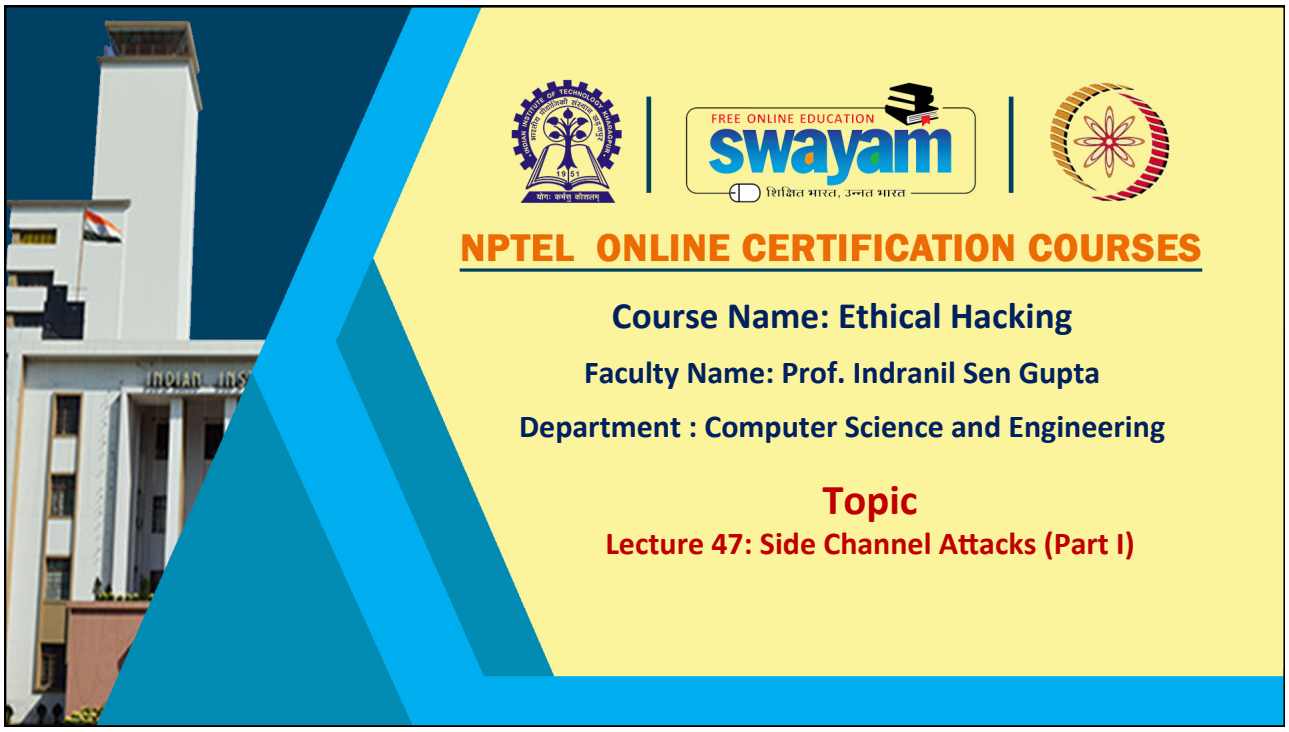

9

To Summarize …

- A hardware implementation of a security device may be based on well-known secure algorithms.
  - The implementation of the hardware may be faulty, resulting in vulnerabilities.
  - The attacker tries to exploit the vulnerabilities.
- Next generation security chips will include countermeasures to protect against such attacks.


10

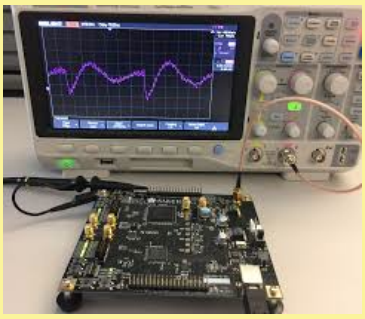

FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Thank you!
11



CONCEPTS COVERED

- Side channel attack
- Timing analysis attack

Side Channel Attacks

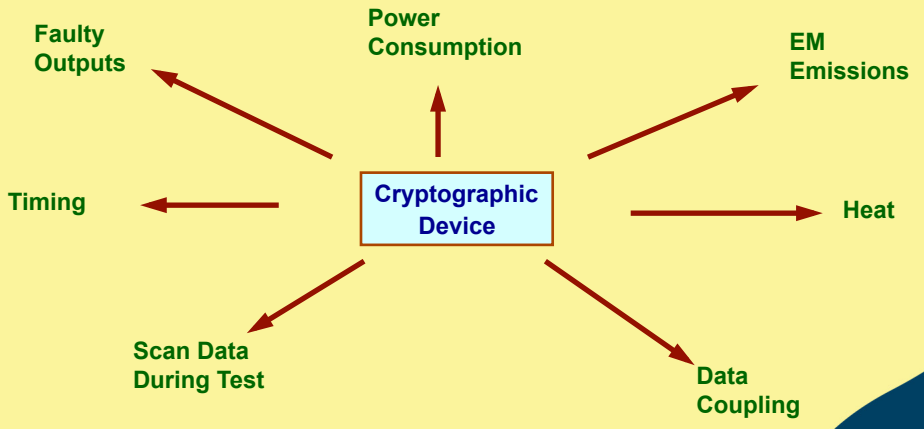


- Side channel attack / cryptanalysis:
  - New research area of applied cryptography.
  - Gained momentum since mid nineties.
  - Basic idea:
    - ❖ Capture unintended leakage of information during operation.
    - ❖ Can be exploited to extract key with relatively low effort.

Why Important?

- A developer of a secure product has to defend it against all possible attack paths.
- In side channel attack:
  - The mathematical security of the cryptographic algorithms is not being questioned.
  - It is the implementation of these algorithms that is at risk to be broken.

Typical Side Channels



5

Timing Analysis Attack

Introduction

- Basic concept
  - The first side-channel based attack to be published [Paul Kocher, 1995].
  - Attacker tries to break a cryptosystem by analyzing the execution time for the overall cryptographic operation.
- What does it try to exploit?
  - Computation time for a private key operation is dependent on the key in some way.
  - Particularly true for asymmetric key algorithms.


7

An Example

- Square-and-multiply algorithm for modular exponentiation (used in RSA, Diffie-Hellman).
  - Execution time depends linearly on the number of '1' bits of the key.
  - Repeated executions with the same key and different inputs can be used.
    - ❖ To perform statistical correlation analysis of timing information.
    - ❖ The key can be recovered completely.


8

- **Square-and-add exponentiation.**
  - How to calculate x^n ?

```
Power (x,n)  =  x,                          if n = 1
             =  Power (x^2, n/2),           if n is even
             =  x . Power (x^2, (n-1)/2),   if n>2 is odd
```

  - **Advantage**:
    - Simple implementation requires *(n-1)* multiplications.
    - This algorithm uses only $O(\log_2 n)$ multiplications.


- Illustration:

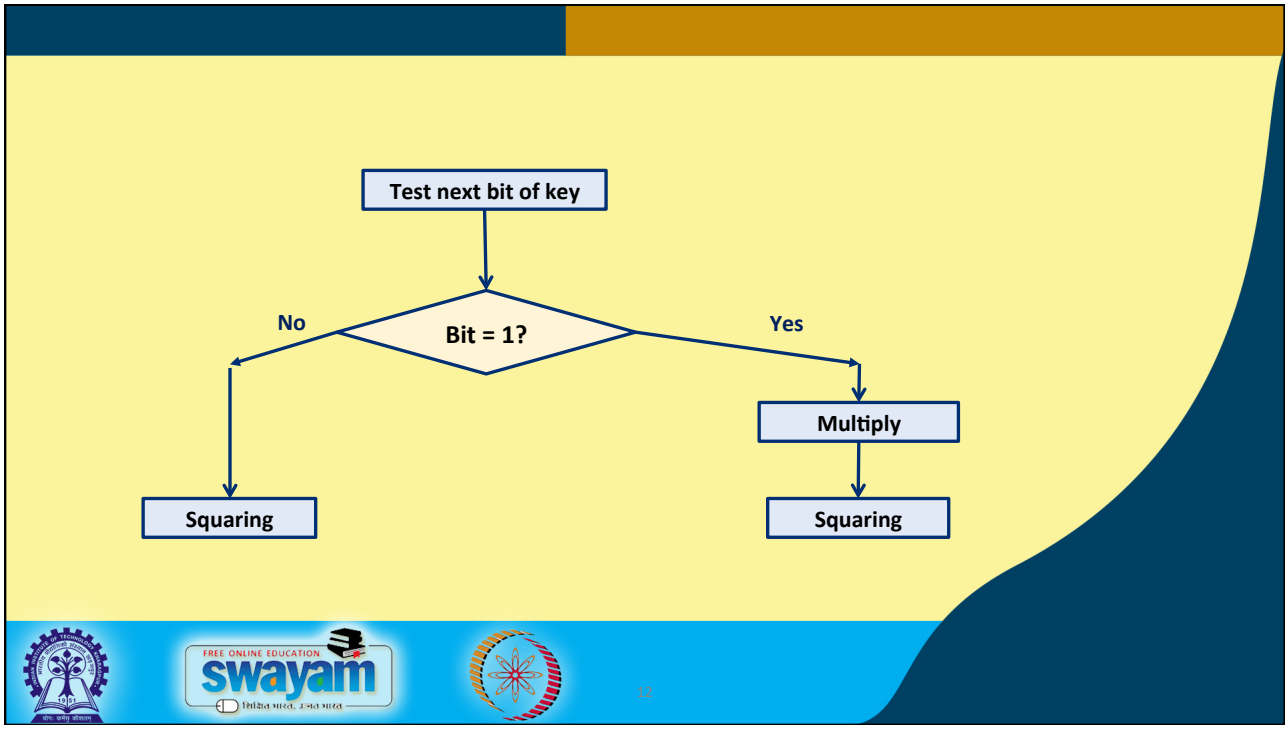
$$x^{13} = x^{1101}$$
$$= x^{(1*2^3 + 1*2^2 + 0*2^1 + 1*2^0)}$$
$$= x^{2^3} * x^{2^2} * 1 * x^{2^0}$$
$$= x^8 * x^4 * x^1$$
$$= (x^4)^2 * (x^2)^2 * x$$
$$= (x^4 * x^2)^2 * x$$
$$= ((x^2)^2 * x^2)^2 * x$$
$$= ((x^2 * x)^2)^2 * x$$

- Requires only 3 squarings and 2 multiplications rather than 12 multiplications.
  - Number of squarings and multiplications can directly give the ***number of 1's*** in the key.


- Pseudo-code to compute $b^e \pmod m$

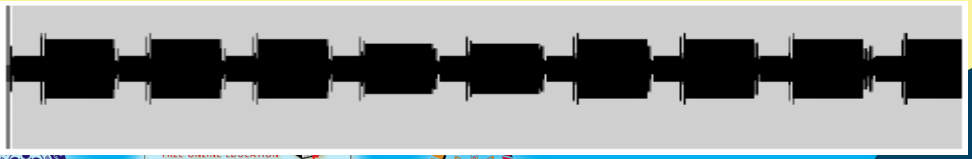
```
Bignum modpow  (Bignum b, Bignum e, Bignum m)  {
   Bignum result = 1;
   while (e > 0) {
     if (e & 1 > 0)  result = (result * b) % m;
     e = e >> 1;
     b = (b * b) % m;
   }
   return result;
}
```

11



- Modified algorithm -- make branches symmetric:

```
Bignum modpow  (Bignum b, Bignum e, Bignum m) {
  Bignum result = 1;
  while (e > 0) {
    if (e & 1 > 0)     result = (result * b) % m;
    else  a = (b * c) % m;
    e = e >> 1;
    b = (b * b) % m;
  }
  return result;
}
```



13

- Timing analysis can reveal the number of 1's in the secret key.

  Time = n * t_{square} + k * t_{mul}

  n : number of bits in the key

  k: number of one bits in the key

  t_{square} : time to compute square

  t_{mul} : time to compute multiplication

- The suggested countermeasure can make the time independent of the key.

  Time = n * (t_{square} + t_{mul})

14

What it means?

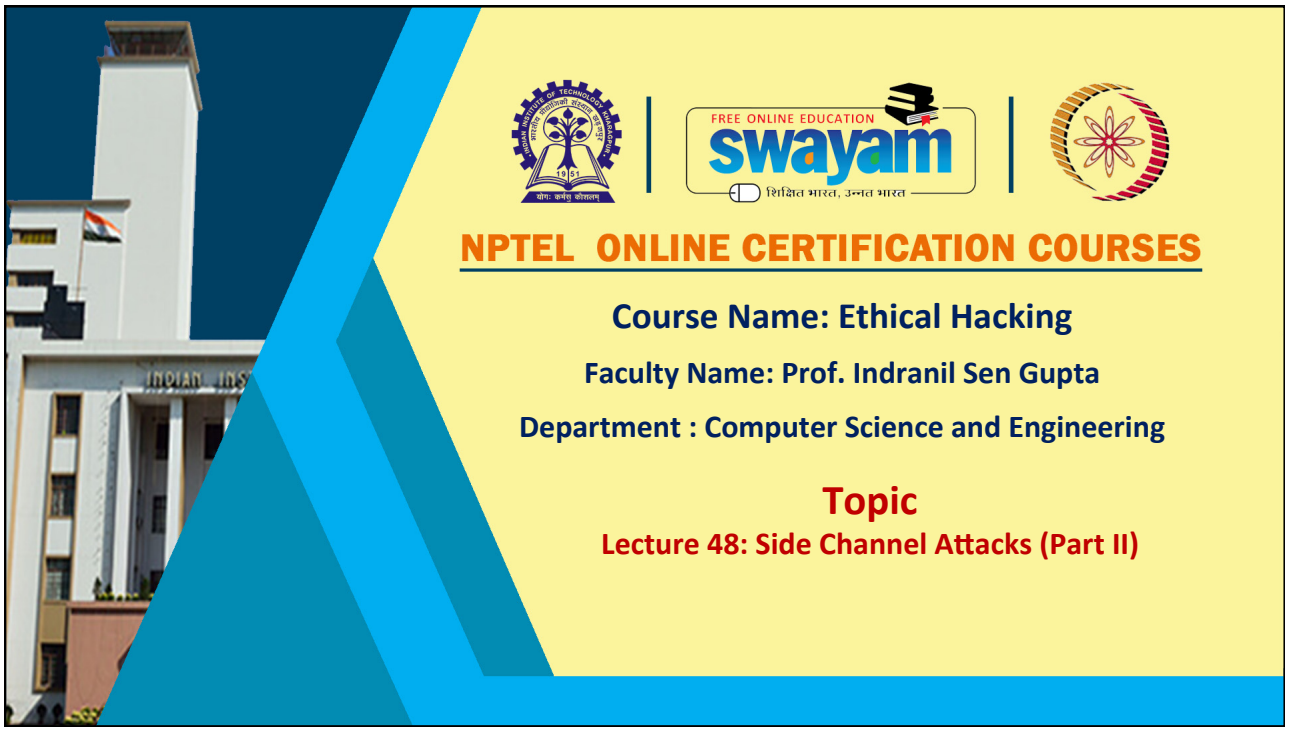
- If the device carrying out the cryptographic operation is available for analysis …
  - We can gain valuable insight into the internal execution.
  - For RSA, the complexity of brute-force attack can be drastically reduced.
- Security implications:
  - We use various sorts of smart cards in our daily life.
  - Side-channel attack can pose a serious threat.
  - Secure side-channel attack resistant implementations are necessary.


15

- An algorithm may be mathematically very secure.
- But weaknesses in hardware or software implementations may make it vulnerable against side-channel attacks.
  - Secure implementation is very important in this context.


16


FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Thank you!
17



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Ethical Hacking

Faculty Name: Prof. Indranil Sen Gupta

Department : Computer Science and Engineering

Topic

Lecture 48: Side Channel Attacks (Part II)

CONCEPTS COVERED

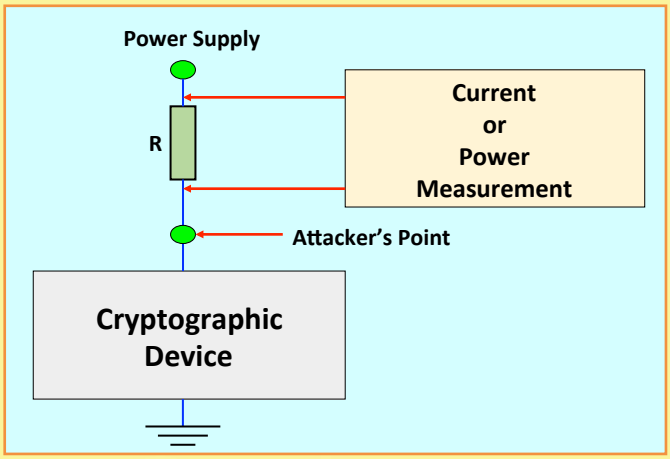
- Power analysis attack
- Simple and differential power analysis
- Countermeasures

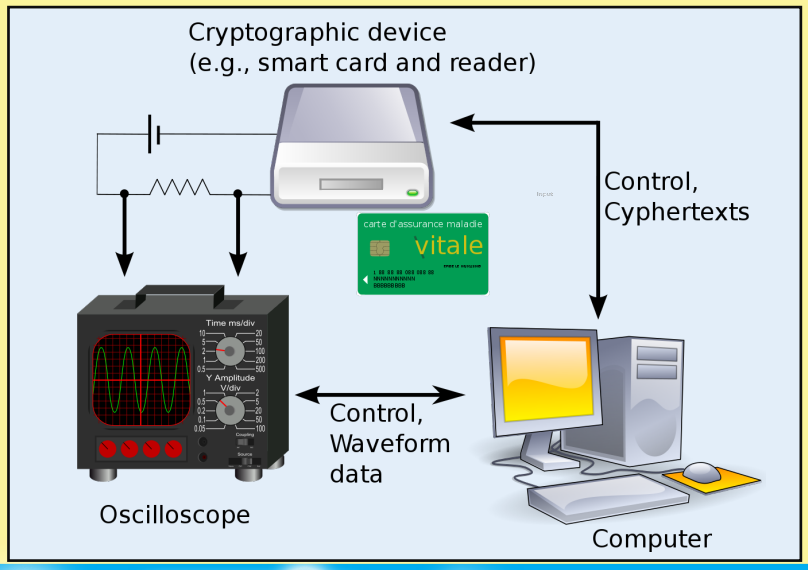
Power Analysis Attack

Introduction

- Basic concept:
  - A much more effective form of side channel attack [Paul Kocher et al, 1998].
  - Analyzes the power consumed by a device during the processing of some cryptographic operation.
- What it can yield?
  - Information about what the device is doing.
    - ❖ Can extract the key information.

Data Acquisition Setup



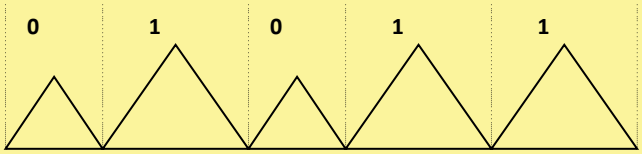


Simple Power Analysis (SPA)

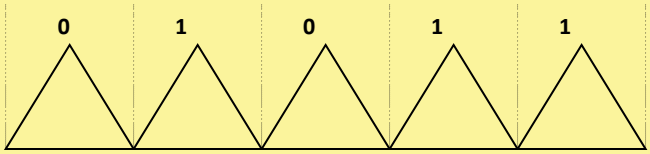
- Attacker directly uses power consumption to learn bits of secret key.
  - Waveforms visually examined.
- Can identify:
  - Big features like rounds of DES/AES, square vs. multiply in RSA exponentiation.
  - Small features, like bit value.
- Relatively easy to defend against.

How SPA Works?

Key = 101011
Square-and-Multiply Algorithm



With "Dummy" Operations

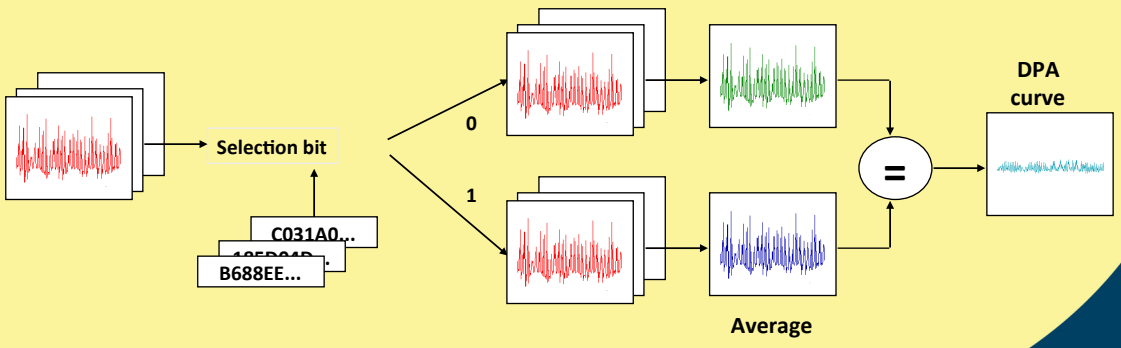


Power Traces

Differential Power Analysis

- More complex.
- Partition the data and related curves into two sets according to selected bits.
- Take the difference, and look for peaks or differences.

DPA Process Schematic



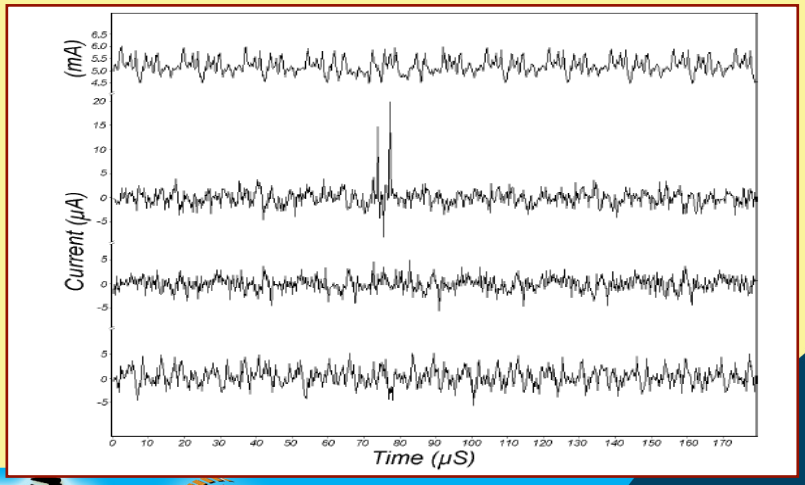
DPA Result Example

Average Power Consumption

Power Consumption Differential Curve With *Correct* Key Guess

Power Consumption Differential Curve With *Incorrect* Key Guess

Power Consumption Differential Curve With *Incorrect* Key Guess



11

Countermeasures

Introduction

- Relatively easy to implement for timing analysis.
  - Make the execution time data independent.
- Power analysis attacks that look at specific intermediate values of the implementation are much harder to defeat.
  - Two broad approaches practiced:
    - **a) Hardware-based**
    - **b) Software-based**



a) Hardware-based countermeasures:

- Special logic styles that minimizes data-dependent leakage.
- Implementation of masking schemes.
- Addition of noise with noise-generators.
- Random process interrupts that provide for an internal timing de-synchronization.



b) Software-based countermeasures:

- Aim to avoid the occurrence of predictable intermediate results.
- Introduce redundant computations.
- Internal randomization used to mask the data representation used.
  - ❖ A random value, not known to the attacker, is added or multiplied with intermediate values.


15

15

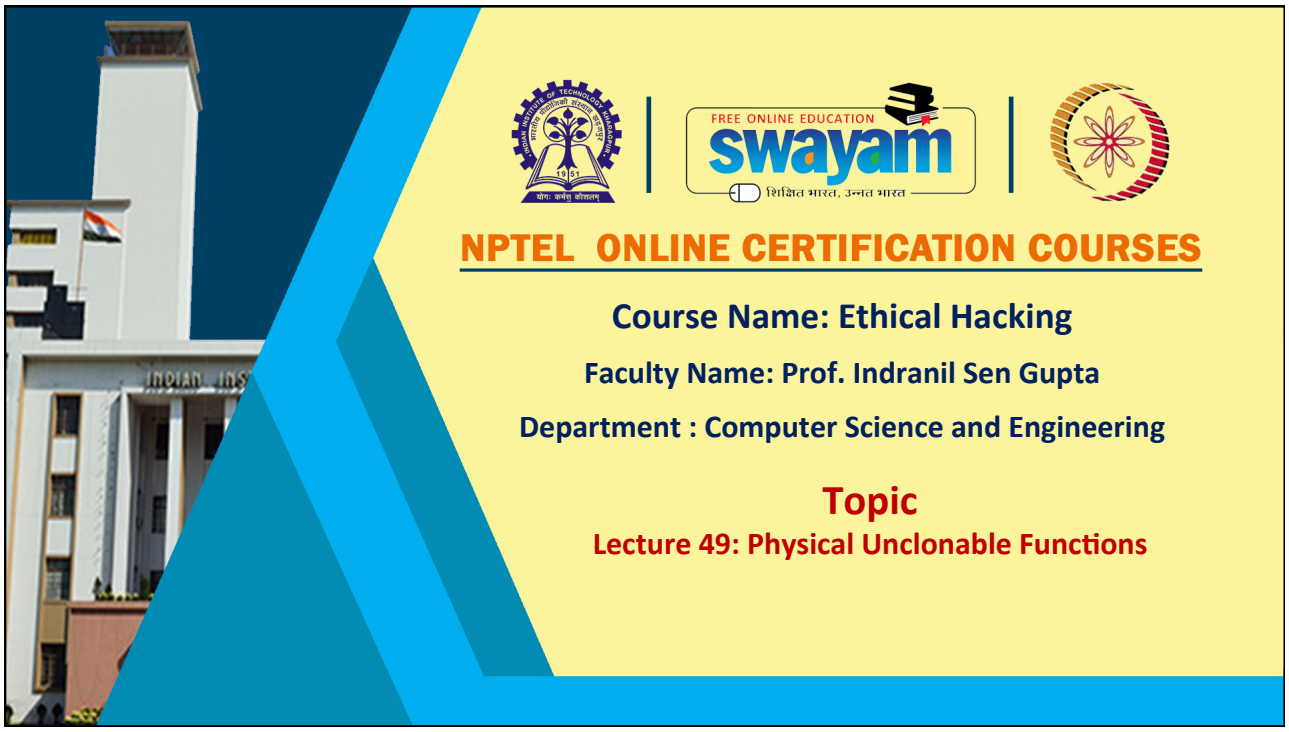
Conclusion

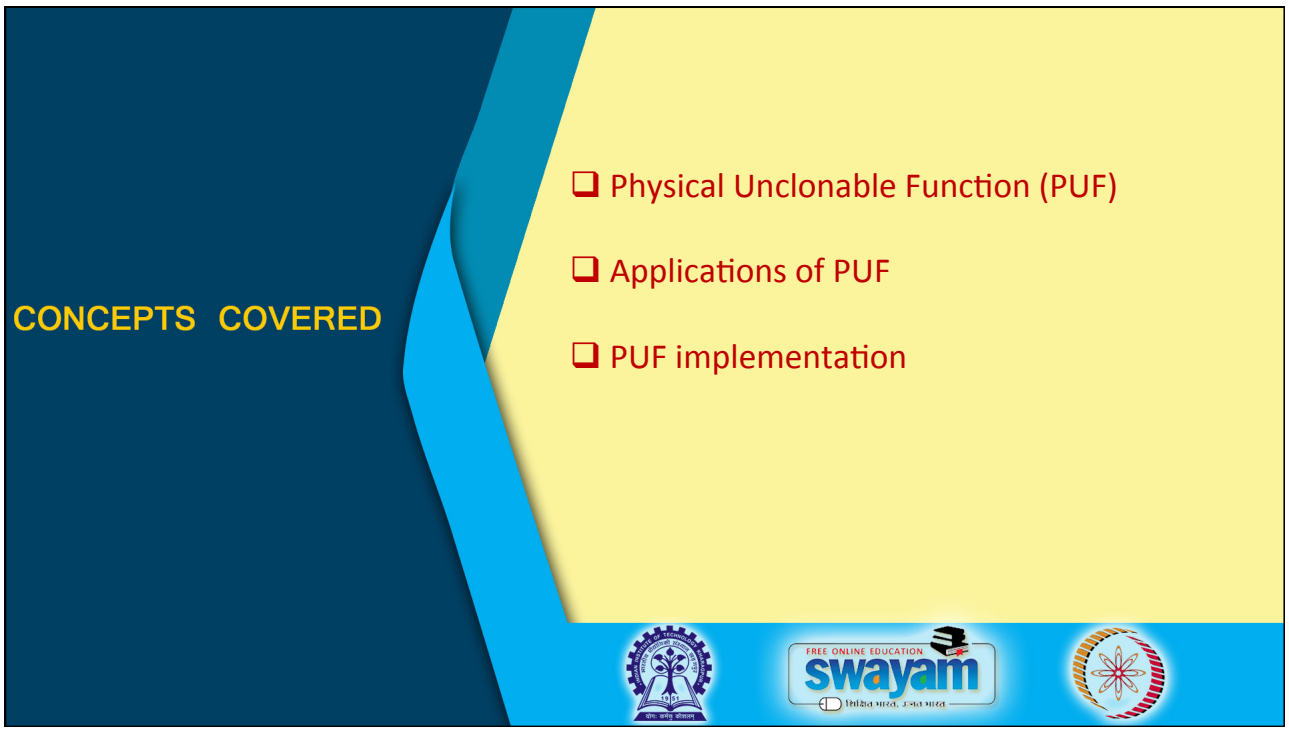
- Side channel attacks
  - Powerful technique but specific.
    - ❖ Targets a particular implementation rather than a generic algorithm.
    - ❖ Most devices as well as software implementations on embedded platforms can be targeted.
    - ❖ Hard to evaluate and prevent.
  - Possible loophole:
    - ❖ Resisting one kind of attack may introduce weaknesses with respect to another one.
  - An active area of research.


16

16

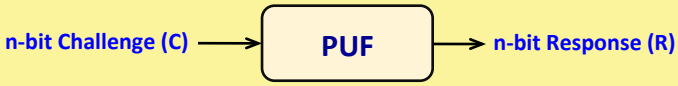

FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Thank you!
17


FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Course Name: Ethical Hacking
Faculty Name: Prof. Indranil Sen Gupta
Department : Computer Science and Engineering
Topic
Lecture 49: Physical Unclonable Functions


CONCEPTS COVERED
❑ Physical Unclonable Function (PUF)
❑ Applications of PUF
❑ PUF implementation
FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत

What is a PUF?

- Fingerprint of some device.
  - A challenge-response mechanism in which the mapping between an applied input ("*challenge*") and the corresponding observed output ("*response*") is dependent on the complex and variable nature of a physical material.
  - The challenge-response mapping is *unclonable* (ideally) and *instance-specific* (depends on manufacturing process variations evident in ASICs).



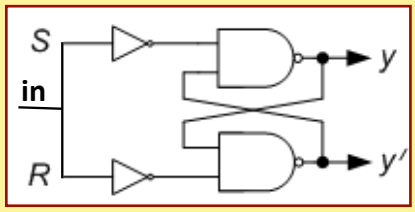
Some Desirable Properties

- **Evaluatable**:
  - Given *PUF* and *x*, it is easy to evaluate *y = PUF(x)*.
- **Unique**:
  - The *PUF(x)* contains some information about the identity of the physical entity embedding the *PUF*.
- **Unclonable**:
  - Given *PUF*, it is hard to construct a procedure *PUF'*, where *PUF ≠ PUF',* and *PUF'(x) = PUF(x)* for all *x*.
- **One-way**:
  - Given only *y* and the corresponding *PUF* instance, it is hard to find *x* such that *PUF(x) = y.*

An Example with a Simple S-R Latch

- Make the input *in=1*.
  - We shall get *y=1, y'=1*.
- Now make the input *in=0*, both of the following states are possible:
  - *a) y=1, y'=0*
  - *b) y=0, y'=1*

Source of randomness

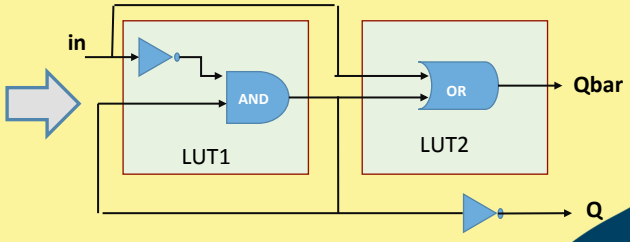


From Theory to Practice

- FPGAs are ideal for security implementations.
  - In-house and high-performance.
  - Programmability is an added feature.
  - But careful implementation is needed.

The non-determinism and hence the randomness is gone!

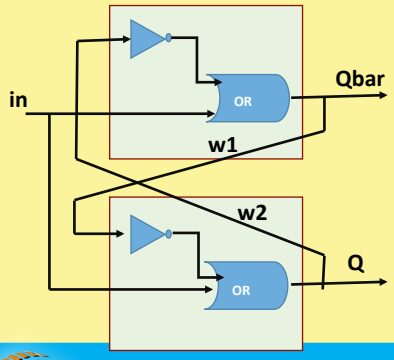
```
module SR (in, Q, Qbar);
  input in;
  output Q, Qbar;
  nand N1(Q, ~in, Qbar);
  nand N2(Qbar, ~in, Q);
endmodule
```



Another Attempt

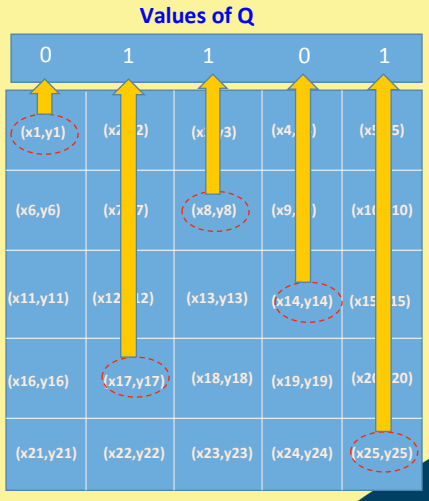
- This design has the non-determinism as expected!
- We can also design using NAND primitives.

```
module SR (in, Q, Qbar);
  input in;
  output Q, Qbar;
  wire w1, w2;
  nand N1(Q, ~in, w1);
  nand N2(Qbar, ~in, w2);
  assign w1 = Qbar;
  assign w2 = Q;
endmodule
```



The Silicon Space

- Mismatch in driving capabilities of the gates.
- Difference in routing delays of the feedback path.
  - A latch cell will give either 0 or 1 as output.
  - Depends on the (x,y) position of the silicon area.



Advantages of PUF

PUF as Security Primitive

- **Reduce Cost**
- **Increase Security**

World without PUF

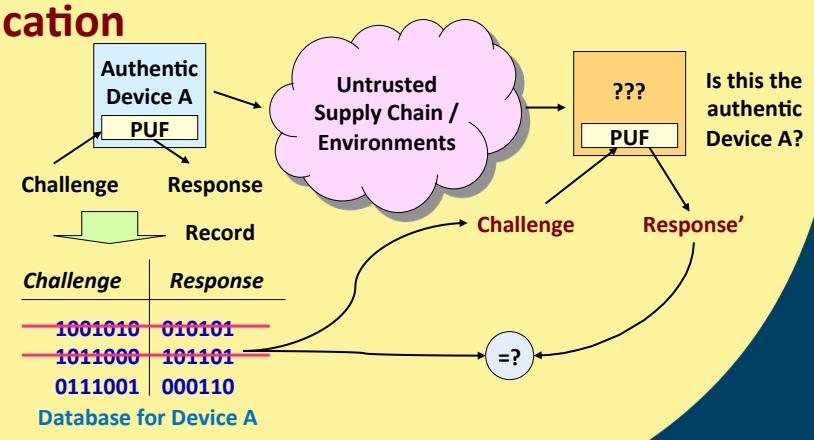
- Trusted party embeds and tests secret keys in a secure non-volatile memory (NVM).
- EEPROM adds additional complexity to manufacturing.
- Adversaries may physically extract secret key from non-volatile memory.

World with PUF

- Intrinsic properties of device is used to generate secret key.
- Key never leaves the IC's cryptographic boundary, nor be stored in a non-volatile memory.
- Key is deleted after usage in encryption or decryption process.

PUFs for Identification

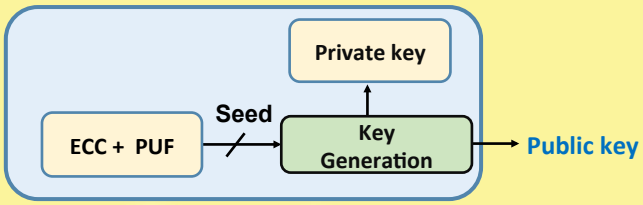
- Protects against ASIC substitution and counterfeits without using cryptographic operations.



IC rejected if *Response'* does not match the enrolled *Response*

PUFs for Private/Public Key Pair Generation

- PUF response is used as random seed to a private/ public key generation algorithm.
- No secret needs to be handled by a manufacturer.
- A device generates a key pair on-chip, and outputs a public key.
- The public key can be endorsed at any time.



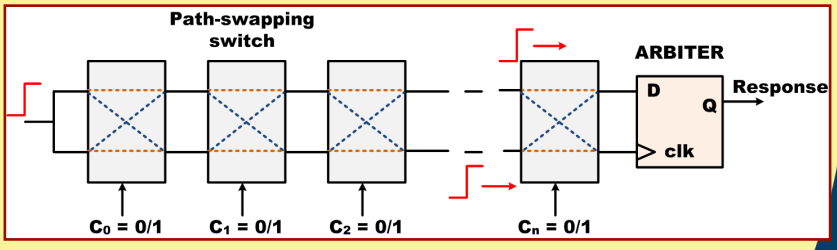
11

Practical Designs of PUF

- We are interested in Silicon-PUF circuits
  - Utilize the unavoidable and inpredictable process variation effects of modern deep-submicron MOSFET devices.
  - From circuit design perspective, process variation is a challenge, but is useful for PUF design.
- Various designs of PUFs have been explored.

12

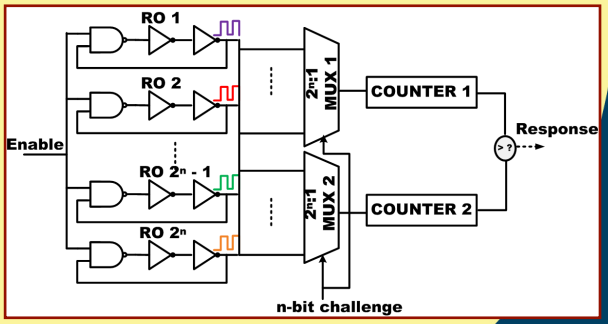
(a) Arbiter PUF



- Composed of *n* two-port switching stages, for an *n*-bit challenge size.
  - Number of possible paths is 2^n.
  - A challenge selects a unique path.
  - Accumulated delay at the end of the path is compared by an arbiter circuit, which gives a 1-bit decision.

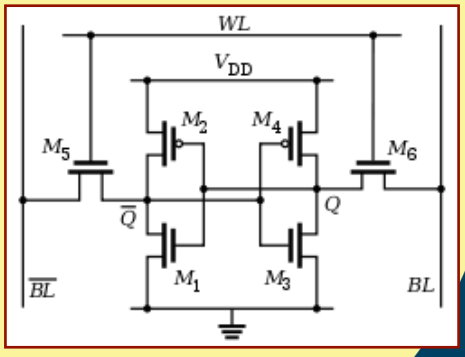
(b) Ring Oscillator PUF (ROPUF)

- An n-bit challenge selects two different ROs from a bank of 2^n ROs.
- Process variation results in ROs to have different oscillation frequencies.
- Compare frequencies of two oscillators using counters.
- A comparator generates the final decision.



(c) SRAM PUF

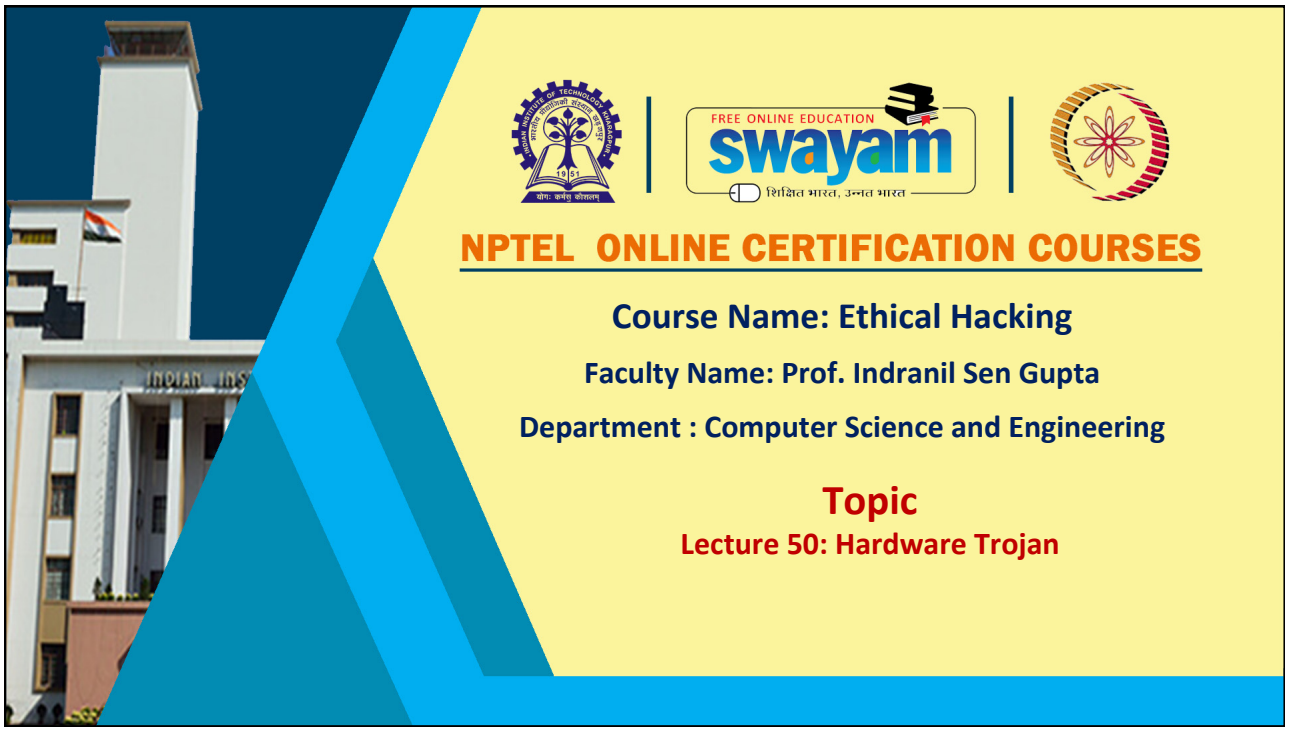
- Power-up initial value of SRAM cell can be used as response; cell address is the challenge.
- SRAM fabrication compatible with digital logic process in regular ICs.
- FPGA implementation of SRAM PUF is very difficult.
  - Since SRAM modules are cleared by default on power-up.



Summary

- PUFs are not very expensive to realize.
- Many recent security protocols are based on PUFs.
  - Makes it difficult to mount hardware-based attacks.


FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Thank you!
17



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Ethical Hacking

Faculty Name: Prof. Indranil Sen Gupta

Department : Computer Science and Engineering

Topic

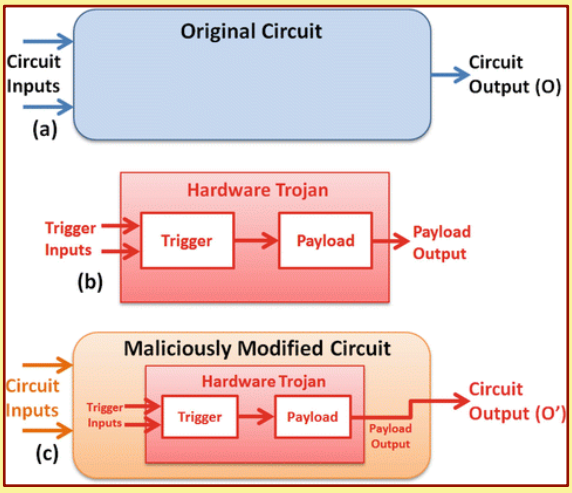
Lecture 50: Hardware Trojan

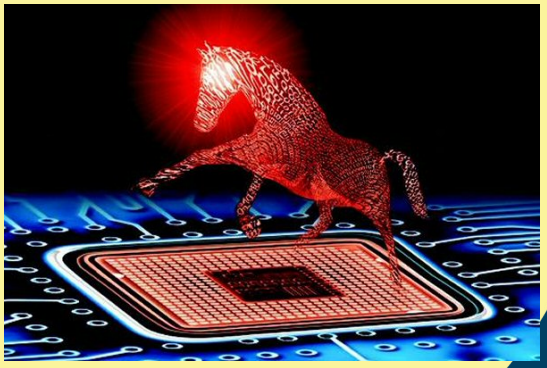
CONCEPTS COVERED

- What is hardware trojan?
- Types of trojans
- Trojan detection

What is Hardware Trojan (HT)?

- It is a malicious modification of the circuitry of an IC chip.
  - During design of fabrication.
- A HT is completely characterized by its physical representation and its behavior.
- The payload of a HT is the entire activity that the trojan executes when it is triggered.



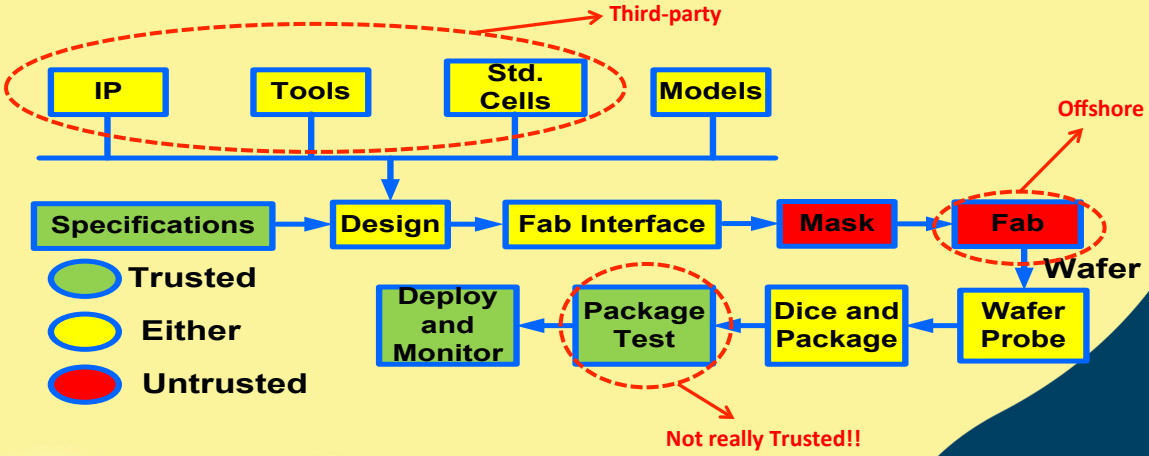


Effects of Prevalent Practices

- Prevalence of IP based design.
- Routine use of CAD tools from EDA vendors.
- Fabless manufacturing model (trend on the rise).
- Outsourcing of manufacturing to offshore fabs.
- Loss of control over design and manufacture.
- Potentially untrusted parties getting involved.


55

Modern IC Design and Manufacturing




6

Hardware Trojans really are …

- **Malicious modifications to design**
  - Can take place pre or post manufacturing.
  - Inserted by intelligent adversary.
  - Extremely small hardware overhead.
  - Stealthy => difficult to detect.
  - Causes IC to malfunction in-field.
- **Results**:
  - Potentially disastrous consequences.
  - Loss of human life and property.

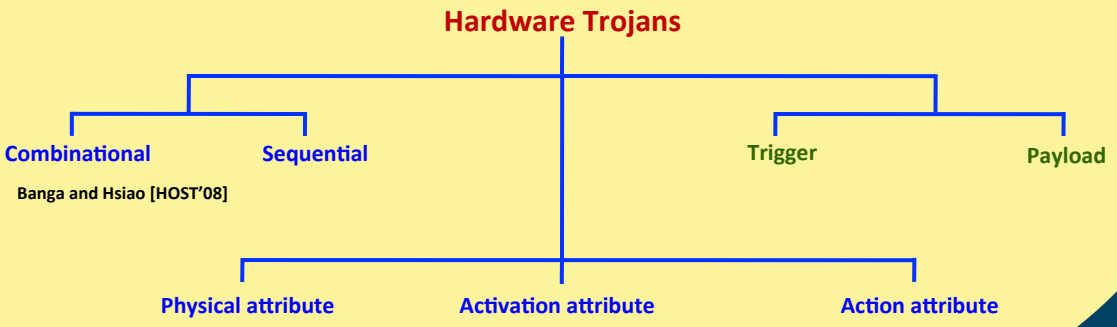

7

How Realistic are Hardware Trojans?

- Do hardware trojans really exist?
  - No concrete proof obtained as yet.
  - Tampering masks in fab is not easy (highly complex).
  - Reverse-engineering a single IC can take months.
- But there is strong evidence they do….
  - Numerous suspected military / commercial cases (as early as 1976!!).
  - Reverse-engineering of ICs is widely believed to be performed by reputed companies (IBM has patents).
  - Highly sophisticated commercial software tools for reverse-engineering are available (Chipworks, etc.), and academic efforts (Cambridge University).

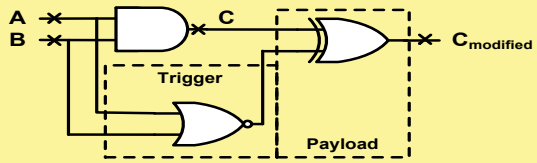

8

Trojan Taxonomy

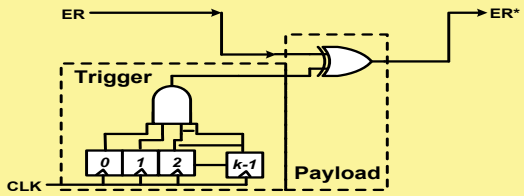


Banga and Hsiao [HOST'08]

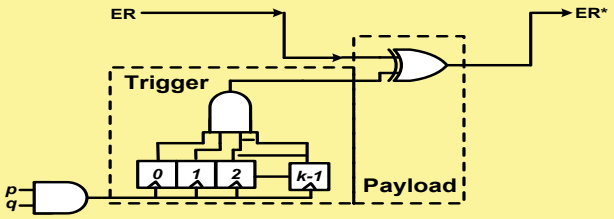
Digital Trojans



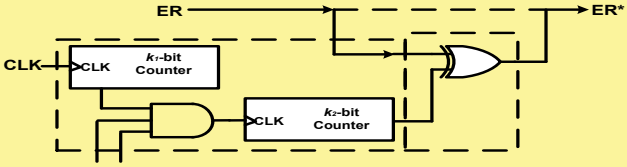
Combinational Trojan (simplest)



Sequential (Synchronous) Trojan ("Time Bomb")

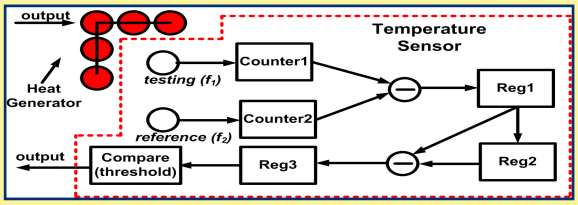


Sequential (Asynchronous) Trojan

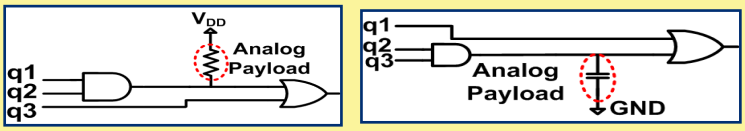


Hybrid Trojans

Analog Trojans

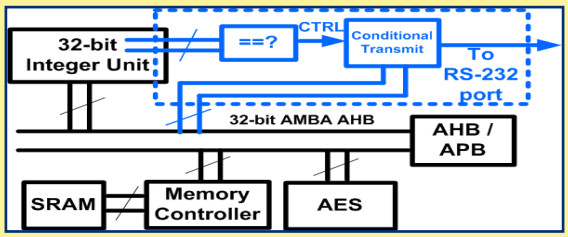


Analog Trojan (activity-triggered)

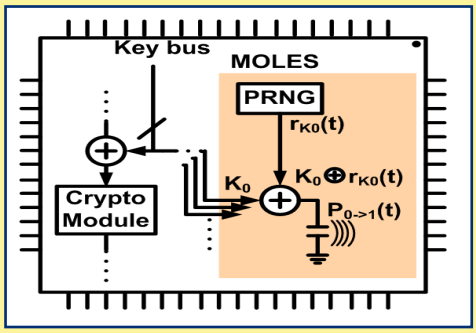


Analog payload Trojan

Information Leakage Trojans

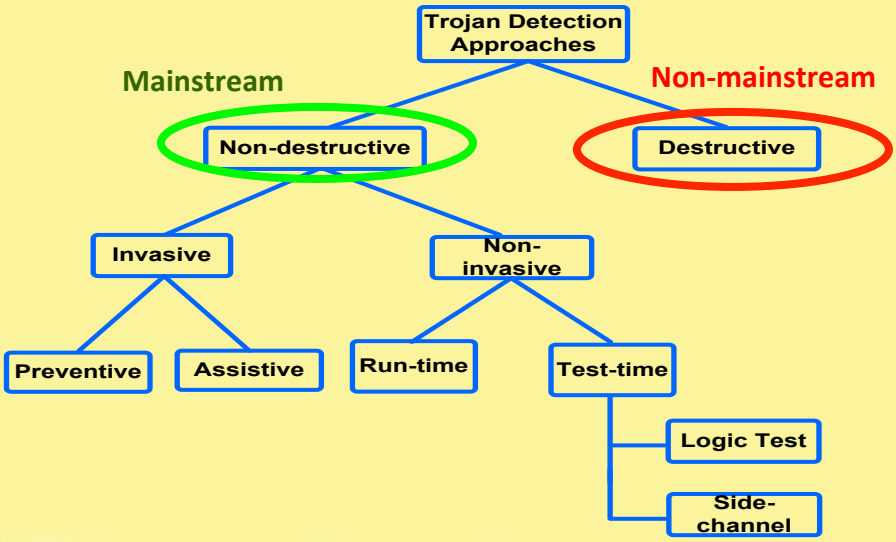


Logic-value Based



Side-channel Leakage Based

Approaches to Trojan Detection



Why is Trojan detection difficult?

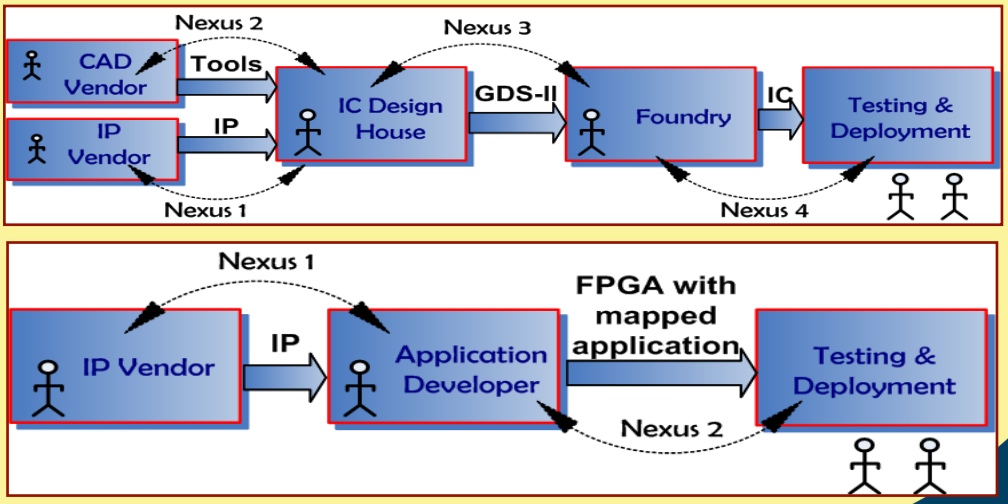
- High overhead.
- Trojans are stealthy.
- Trigger nodes have very low observability.
- Extremely large number of Trojan types possible.
- No single method can detect all types of Trojans.


15

Multi-Level Attack

- Uses nexus between multiple parties.
- Only parties which are part of the nexus can benefit.
- The nexus eases the burden on individual parties.
- Additional challenges to detect.


16



Conclusion

- IC design/manufacturing practices are insecure.
  - Third-party IPs and off-shore manufacturing.
  - Potentially *untrusted* parties pay a major role.
- Hardware Trojans are malicious circuit modifications.
  - Small overhead, hugely destructive impact.
  - Difficult to detect by traditional testing means.
- State-of-the-art:
  - Both design and test techniques have been proposed.
  - Effectiveness of the proposed techniques limited to the particular types of Trojans.


FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत
NPTEL ONLINE CERTIFICATION COURSES
Thank you!
19