Hardware Security in IoT

Introduction

- The extremely complex IoT ecosystem involves a large number of interconnected IoT devices, and this number is expected to significantly increase within a few years.
- The security of the IoT environment must be given a high priority while developing, configuring, and updating the devices to ensure optimal performance throughout their lifetime.
- Protection of IoT devices against cyber-attacks can help achieve a high level of confidentiality, integrity, and availability with authorized access to the users.
- Security of IoT devices is a work in progress, and proliferation of these devices at an alarming rate is making it more challenging to secure them.
- IoTs are an attractive target for adversaries because of their easy accessibility, vulnerabilities, and the quality of data they hold.
- IoT devices are usually connected to other devices on the same network, which puts all other devices at risk if one of them gets compromised.
- Therefore, security of these devices regardless of their size is paramount

Challenges

A lot of IoT devices are extremely small in size and fall into the category of low-resource devices (LRD) or constrained-resource devices (CRD). Their physical limitation allows them to possess low-power resources only, which makes adding more layers of security difficult. Constrained resources, such as limited central processing unit (CPU) and memory, limit their ability to process information and complex security algorithms. There is a serious tradeo between security and resources when it comes to IoTs. Confidentiality is most significantly sacrificed due to constraints in size and power of IoTs

Public Key Infrastructure (PKI)

- Implementation of Public Key Infrastructure (PKI) can provide confidentiality and integrity.
- However, the encryption process with public key demands computational and memory resources that are beyond the capabilities of many IoT devices.
- Also, PKI requires the devices to be continuously updated as the certificates expire. Updating IoT devices is another challenge, especially in really small devices which do not have any user interface.
- Moreover, some IoT devices are deployed remotely in inaccessible areas and cannot be updated without human intervention

Hardware Security Managers (HSM)

- Attestation along with Hardware Security Managers (HSM) is used to achieve integrity.
- HSM provides security of authentication keys.
- These keys are usually encrypted with a human's password or another identification parameter, and thus require human intervention.
- This is difficult to implement in cases of inaccessible and unattended IoT devices.
- Thus, another method needs to be implemented to protect authentication keys.
- An alternative way to protect the keys could be storing them in a dedicated hardware storage. However, firmware modification of the device could lead to authentication keys being read by modified firmware, and they might fall into hands of an adversary.

Authentication and identification

- Authentication and identification of users in IoTs is also significantly important.

- If an attacker compromises authentication, they can get access to the system as a legitimate user and can launch various further attacks without even being detected.

- Current authentication methods include:
  - username/password, digital certificates, shared keys, biometric credentials.

- It is anticipated that IoTs as pervasive will remove many physical interaction interface mediums through which username/passwords are passed.

- Furthermore, verifying identity can be challenging in the mobile environment of multiple IoT devices. Different users move their IoT devices through Different architectures and infrastructures provided by Different service providers.

Effective Access Control

- To attain effective access control is another challenge in the IoT domain.
- It is difficult to use well-known access-control models such as role-based access control (RBAC) and attribute-based access control (ABAC) for low-powered devices [9].
- Furthermore, access control requires the concept of authentication and identity, which is already a challenge in IoTs, as discussed earlier.
- The resource-constrained nature of IoTs could lead to energy consumption and resource exhaustion attacks, which can lead to Denial-of-Service (DoS) attacks, affecting availability as well.
- Nonrepudiation is another challenge in IoTs, which cannot be achieved without proper attestation.
- Proper attestation is already hard to achieve in IoTs.

Hardware threats

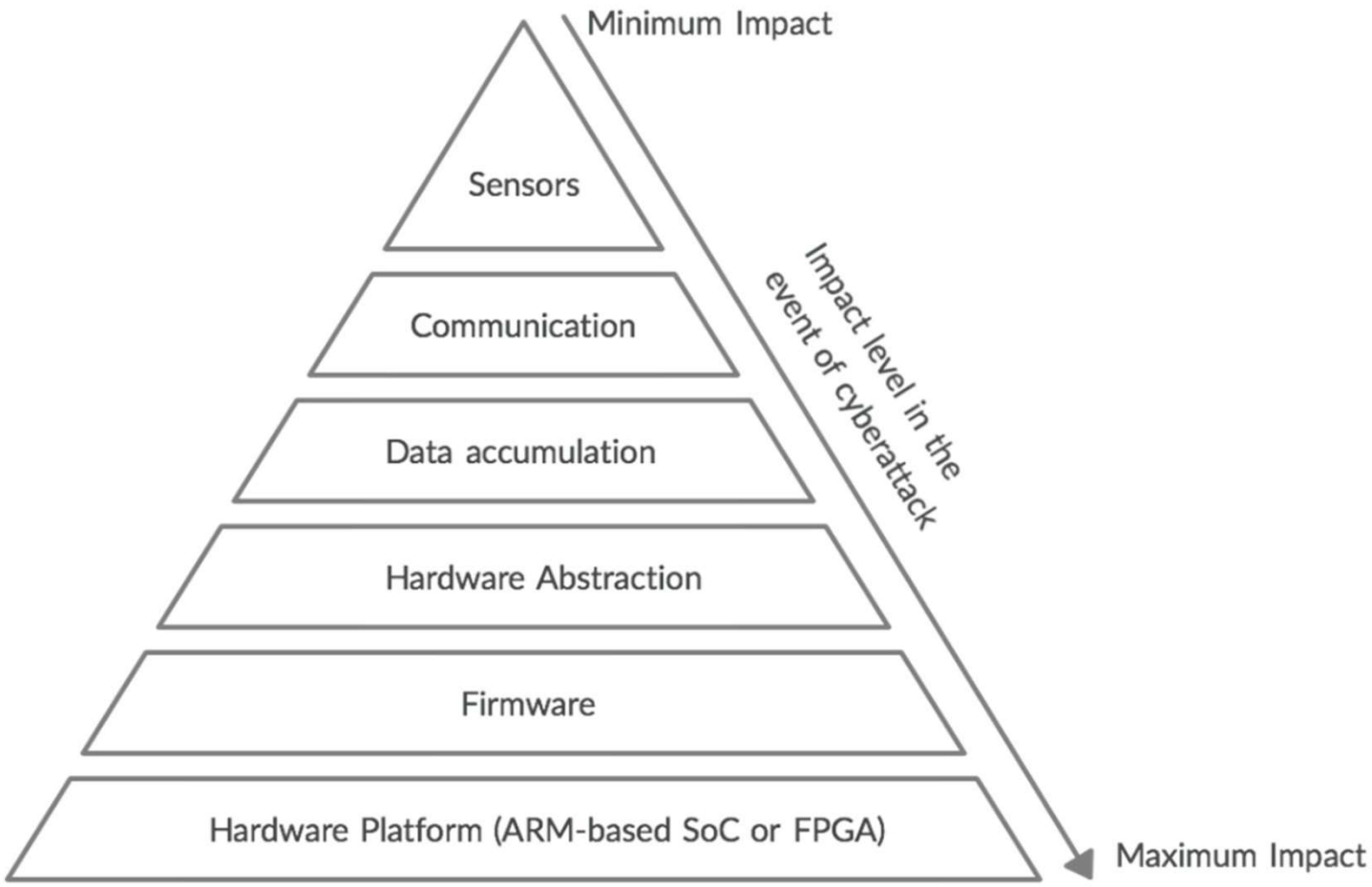
- Hardware threats are increasing at a higher rate than the implementation of hardware security controls.

- Some IoT devices are deployed in easily accessible, unfriendly, and less supervised areas, which allows attackers to physically get a hold of the object and tamper it.

- Hardware vulnerabilities are very difficult to detect.

- An example of it is the attack suffered by Ivy Bridge intel processors, that included transistors of chips being doped to change the random number generator (RNG). RNG is an important base for encryption systems, and the attack resulted in a fixed output of RNG. Detecting such a small modification in a circuit is very challenging.

Pyramid of Pain (based on vulnerability analysis)



Hardware Trojan (HT)

- An HT is a deliberate malicious insertion or alteration to the existing circuit, resulting in change of functionality of the circuit when activated.
- HT can infect any kind of IC, such as application-specific integrated circuits (ASIC), system-on-chip (SoC), field-programmable gate arrays (FPGAs), and digital signal processors (DSP).
- Unlike software Trojans, HTs cannot be eliminated simply by a firmware update.
- HTs may cause many damaging effects to the IoT ecosystem, such as information leakage, denial of service (DoS) attacks,service degradation, and failure of the device.

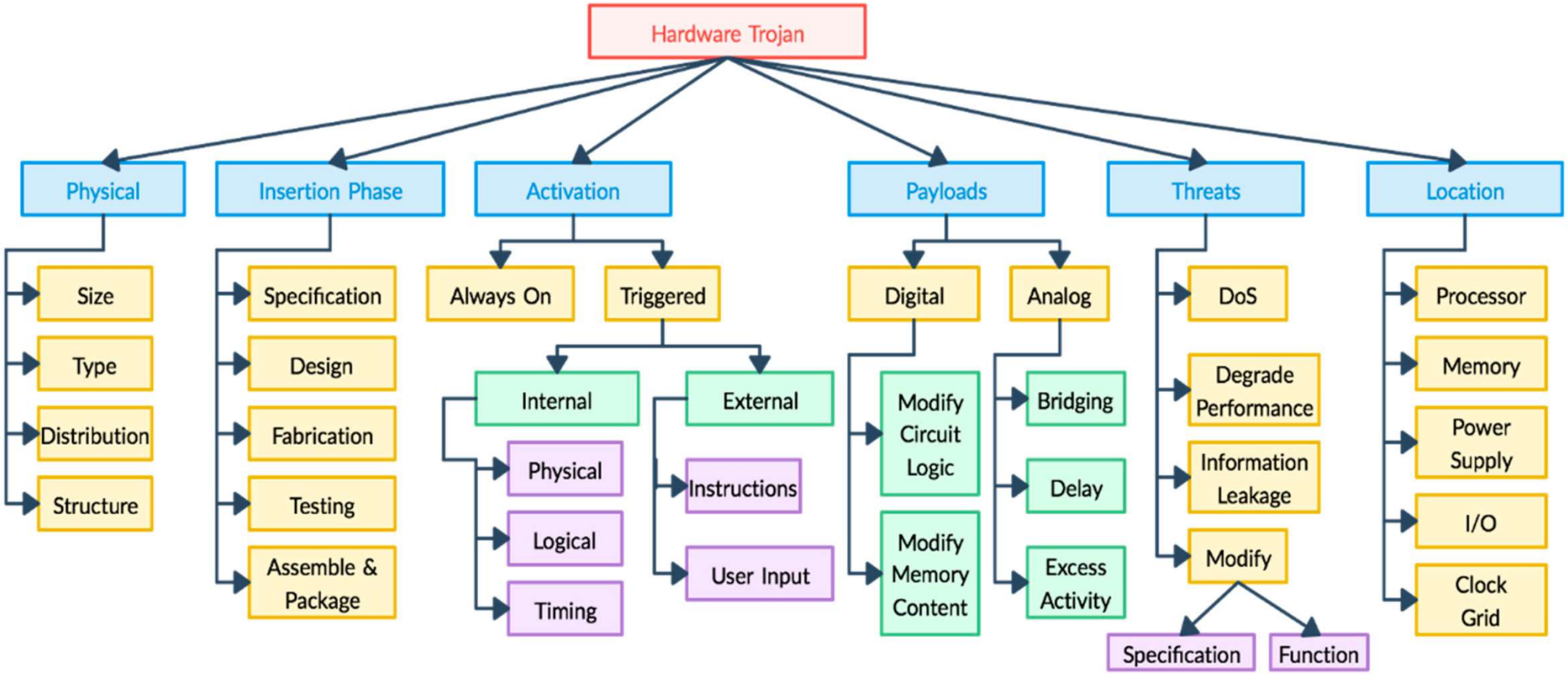
Hardware Trojan Taxonomy

- To grasp the understanding of HTs, it is important to discuss its taxonomy.

- Proper classification of hardware trojans forms a good basis for the implementation of appropriate solution techniques.

- HTs based on the activation property of Trojans:
  - Always On
  - Triggered(internally triggered or externally triggered).

Hardware Trojan Taxonomy



Hardware Trojan Taxonomy

Physical: First, classification of HT taxonomy is based on physical attributes of the trojan, which is further distributed into size, type, distribution, and structure.

- **Type**: Type distributes Trojan into functional and parametric classes. Trojans that are introduced through addition or removal of gates or transistors fall into the functional category, whereas Trojans introduced by modifying existing wires or logic belong to the parametric category.
- **Size**: Size of a trojan depends on the number of components in the chip added or deleted. The smaller the trojan, the higher its probability for activation.
- **Distribution**: This signifies the location of the Trojan in the chip. If a Trojan's components are topologically close in physical layout of the chip, it is categorized into tight distribution. If a Trojan is dispersed across the layout of the chip, it falls into the loose distribution class.
- **Structure**: Adversaries try to make sure that insertion of a Trojan should not change the physical layout of the circuit in order to evade detection.

Hardware Trojan Taxonomy

Insertion Phase: This signifies the stage at which the Trojan may be inserted during design and manufacture of an IC. The various stages that offer an entry point for HT insertion are specification, design, fabrication, testing, and assembly.

Activation: HTs can be activated by the occurrence of certain events inside or outside of the system.

- However, some Trojans do not require a trigger to be activated and can disrupt the chip's function any time. Such Trojans fall into **Always On category**.
- Others require an **internal or external trigger** in order to be activated.
- The internal trigger could sense environment/conditions around the device, being caused by a physical sensor on the IoT device that detects things such as humidity, temperature, etc. An internal logical state or counter value may also activate a Trojan.
- Some Trojans are designed to be **activated** at a certain time or after a specific **amount of delay**. The external triggers include instructions sent remotely after taking advantage of weak network security or an input entered by the user without the user's knowledge.

Hardware Trojan Taxonomy

- **Payload**: A payload is the information that links a undesirable event with activation of the Trojan.

- Once the expected condition is detected by the trigger, the payload is activated, and Trojan starts performing the malicious activities.

- The payload mechanism of an HT is classified into digital and analog.
  - Digital Trojans can modify logic values at specific internal payload nodes, or they can modify memory contents as well.
  - Trojans with an analog payload can affect parameters of the circuit such as performance, power, noise, etc.

Hardware Trojan Taxonomy

Threats: The classification of HTs can be done based on the threats they pose.

- HTs can lead to Denial of Service (DoS) attacks, performance degradation, and information leakage.

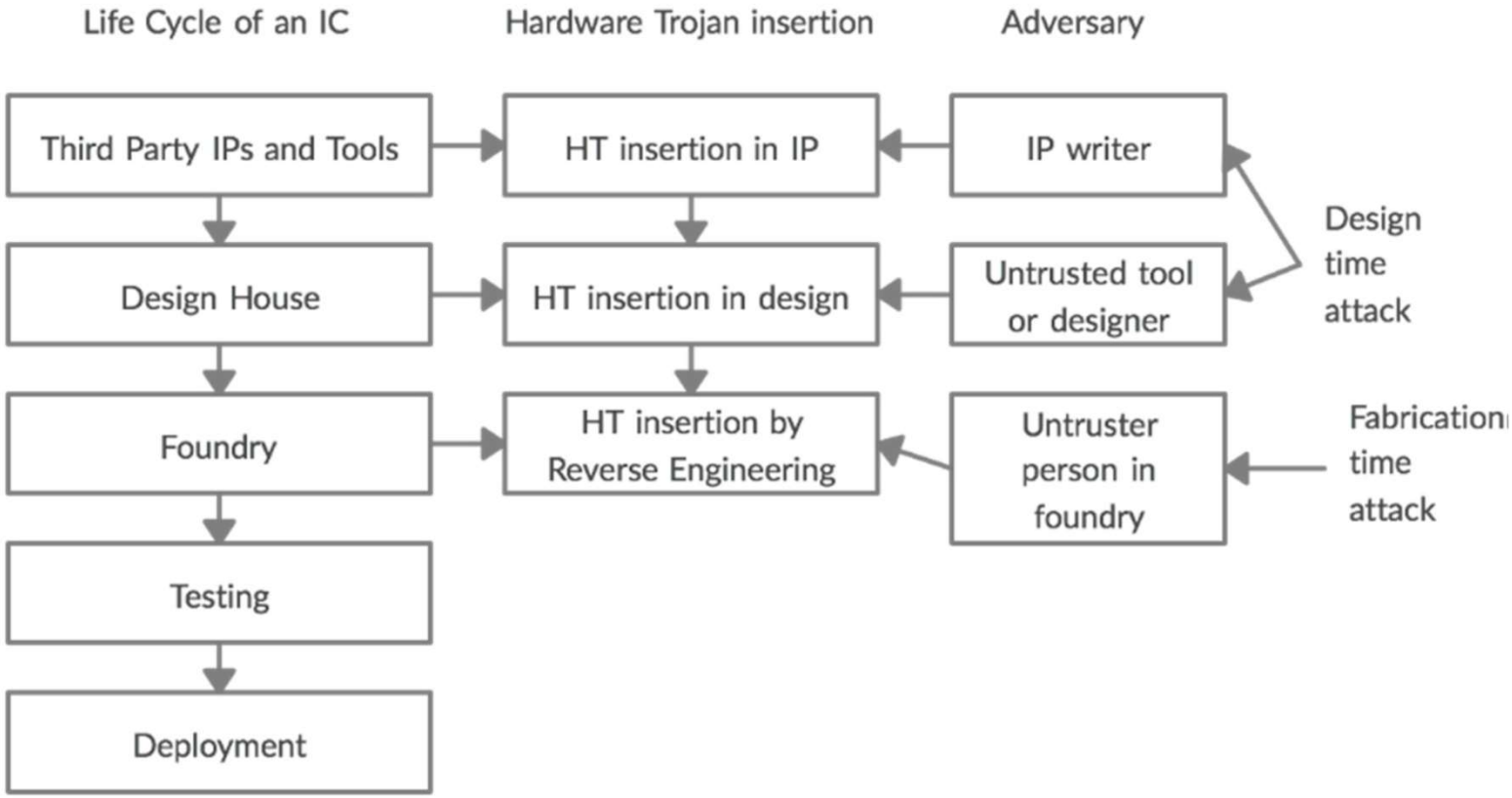
- Other threats that HT present are modification of function of the chip by adding or removing logic, or modification of specifications of the chip, which refers to alteration of the parametric properties of the chip such as delay.

Location: This refers to the physical location of the Trojan in the circuit.

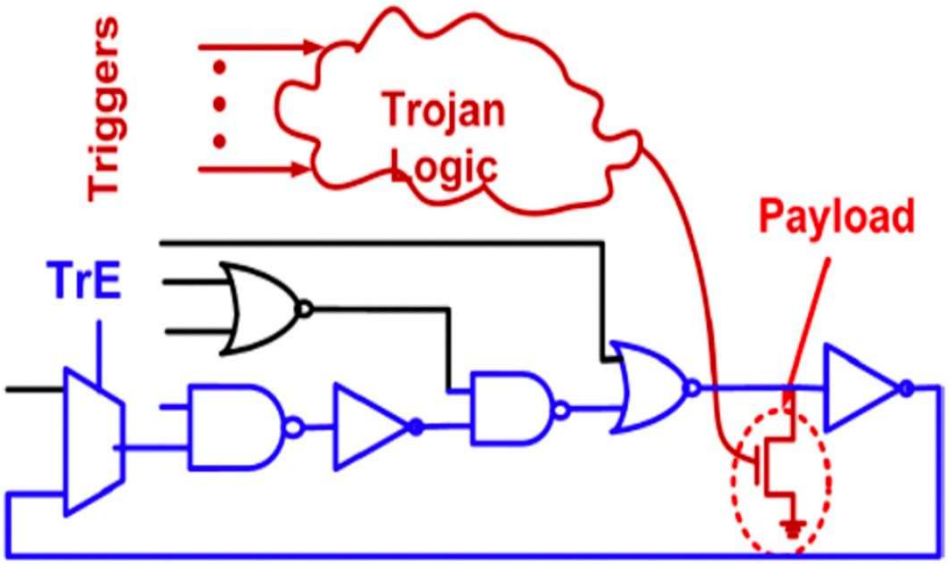
- The locations where HTs can be present are processor, memory, power supple, Input/Output (I/O), and clock grid.

Hardware Trojan Insertion

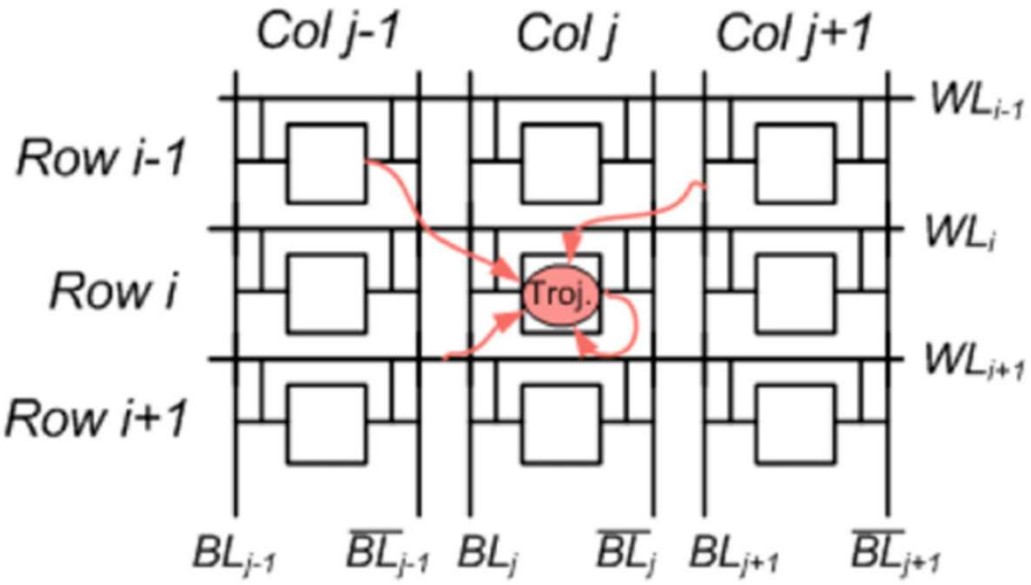
Globalization and outsourcing of chip manufacturing open the gates for HT insertions at any point during the fabrication or design process, either in untrusted foundries or in design houses.



Hardware Trojan Insertion



Payload insertion by-passing transistor payload



Hardware Trojan insertion in SRAM (Static Random Access Memory) array

Untrusted parties

Table indicates each model, with the untrusted party involved in the model. The models consider three parties involved in the supply chain: third-party IP (3PIP) vendor, foundry, and SoC developer

Trojan Models based on untrusted parties

Model Number	Trojan Model	Untrusted Parties
A	Untrusted 3PIP (Third-party IP)	3PIP vendor
B	Untrusted fabless design house	Foundry
C	Untrusted SoC (System on a Chip) developer	SoC developer
D	Untrusted commercial off-the shelf (COTS)	3PIP vendor, foundry and SoC developer
E	Untrusted design	3PIP vendor and SoC developer
F	Untrusted outsourcer	3PIP vendor and foundry
G	Untrusted system integrator & foundry	System integrator and foundry

Side-Channel Attacks

- Side-channel attacks (SCA) are noninvasive hardware-based attacks in which adversaries use forensic techniques related to the physical implementation of the embedded hardware to extract information.

- Side-channel information such as power, electromagnetic radiations, timing information, or even sound can be analyzed to implement this attack. This information gathered from a system without accessing the system directly coupled with appropriate calculations can help the attacker retrieve the secret cryptographic key.

- Researcher proposes a wireless interceptive SCA technique by performing a Correlation Electromagnetic Analysis (CEMA) attack against an AES-128 encryption algorithm for IoT applications, and successfully reveals the secret key.

Side-channel attacks are often classified into two categories:

Active side-channel attacks
Active attacks involve exploitation of the side-channel inputs where the device's functioning is tampered, such as fault-injection attacks in which error is induced in the computation.

Passive side-channel attacks
Passive side-channel attacks are the ones that exploit side-channel output, where an adversary simply observes the behavior of the device without disturbing it, such as in an electromagnetic analysis attack, which is described later.

Attackers require special and expensive equipment such as probes, an oscilloscope, a bandwidth amplifier, analyzing software, etc

Acoustic cryptanalysis key extraction attack:

- Researchers demonstrates how such an attack can extract full 4096-bit RSA decryption keys from laptop computers (of various models), within an hour, using the sound generated by the computer during the decryption of some chosen ciphertexts.
- The research experimentally determined that such attacks could be carried out using either a plain mobile phone placed next to the computer or a more sensitive microphone placed 4 m away.

Electromagnetic Analysis Attacks (EMA):

- This attack is based on the analysis of captured electromagnetic radiation from a hardware device.
- These emanations could be captured from a distance and used to reconstruct the information being displayed on a computer monitor.
- These attacks have also been performed on integrated chips (IC) by placing tiny antennas close to the victim EMA attack shows an image with a computer display reconstructed from electromagnetic radiations of a computer screen.

Power Monitoring Attacks:

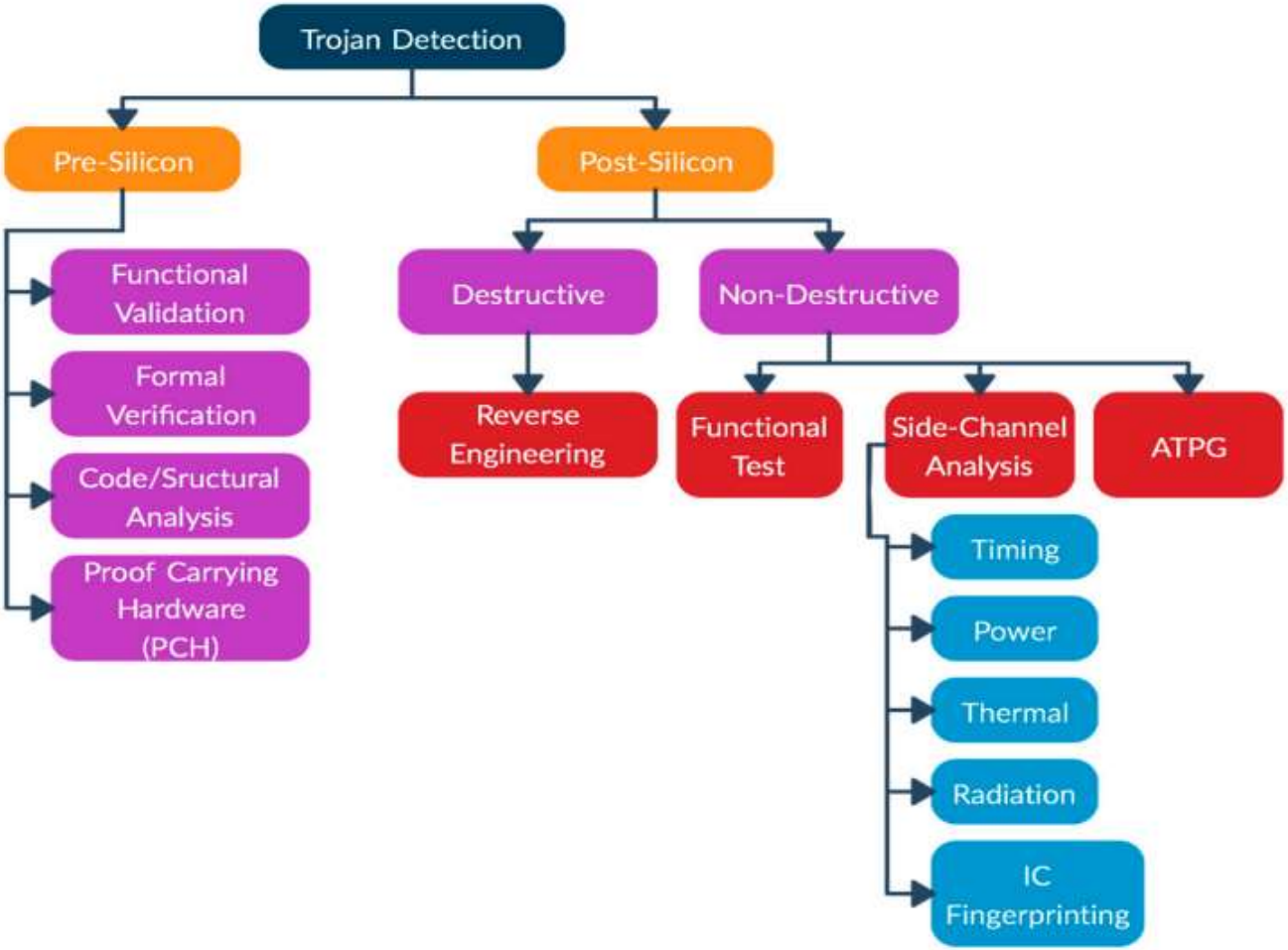
- In such attacks the adversary analyzes varying power consumption by a hardware device. These attacks have the capability of extracting cryptographic keys from the device.

- Simple power analysis (SPA) is straightforward and uses visual interpretation of power traces. However, differential power analysis (DPA) is a more advanced attack that is based on an analysis of measurements of power levels at different parts of the chip in addition to statistical analysis to overcome countermeasures such as added noise.

- The analysis helps the adversary identify computational operations being done by a device and reveals several bits of the crypto-key at a time.

- Full key is derived after a few repetitions of the whole process

Trojan Detection Countermeasures



Trojan Detection Countermeasures

Postsilicon Trojan detection techniques:

These techniques can be classified into destructive techniques and nondestructive techniques.

- **Destructive Techniques**: In destructive techniques, reverse-engineering is used, which involves de-packaging an IC, and design-for-trust validation of the product is reconstructed from the obtained images of each layer.
- This method provides 100% detection rate of any malicious modification in the IC, but it is very high-cost and time consuming.
- Also, the chip cannot be used after the process, and information of only that particular chip is gained.
- This method is good to perform on a few samples to obtain the golden IC model's characteristics. Golden chip is a trojan-free, perfectly functioning IC that is used to be matched against other ICs.

Trojan Detection Countermeasures

Nondestructive Technique: These techniques are used to verify fabricated ICs from untrusted foundry using the following methods:

Functional test:

- These tests require the activation of Trojans by using test vectors, and then the responses are compared with the correct results.
- The stealthy trojans gone undetected during manufacturing test process can be detected during these tests.
- The Trojans that do not change the functionality of the original circuit may not get detected by the functional tests.
- In order to hide from accidental triggering and side-channel analysis, HTs try to choose inactive nets (nets that rarely switch) in order to keep power leakage or extreme nets (nets that spend most time on one state) to a minimum.
- Circuit-under-test (CUT) can be evaluated, and inactive nets can be found by calculating the probability of switching of all the nets.

Trojan Detection Countermeasures

Nondestructive Technique: These techniques are used to verify fabricated ICs from untrusted foundry using the following methods:

Side-channel analysis

- This method takes advantage of side effects produced by additional circuits or Trojan activation, i.e., extra path delay, power, heat, etc., which are measured to detect HTs.

- An IC fingerprinting technique uses side-channel information such as power, temperature, EM, etc., to construct fingerprints with noise modeling for an IC family.

- Most of these techniques require comparison against golden ICs, which is not always available. Also, highly sensitive instruments are needed to measure small side-channel signals such as leakage current, as Trojans require very little current due to their size.

- HT detection method based on side-channel analysis which shows that measuring path delay on 20 paths can help detect more than 80% of HTs.

Trojan Detection Countermeasures

Nondestructive Technique: These techniques are used to verify fabricated ICs from untrusted foundry using the following methods:

ATPG (automatic test-pattern generation) method:

- This method uses the application of digital stimulus to the chip and analyzes the digital output.

- One scheme called MERO (Multiple Excitation of Rare Occurrence) uses a statistical approach based on generating test vectors, which excite rare nodes simultaneously and increase the probability of an HT being triggered and detected easily.

- The research proposes the use of this technique with side-channel detection techniques to increase their impact. Another ATPG-based research discusses to change the design rules to detect HT that is inserted in the chip's existing logic.

Trojan Detection Countermeasures

Presilicon Trojan detection technique: SoC developers and design engineers use these techniques to validate 3PIP cores and their designs. The presilicon detection techniques can be classified into the following three classes

Functional validation: This technique uses the same principle notion mentioned in the functional test described under post-silicon techniques. Functional tests are performed on a tester, which involves collecting output responses to each input pattern provided, but functional validation is conducted with simulation using existing functional testing techniques

Code/Structural analysis: Behavioral or structural codes are tested to detect any redundant statements or circuits that might be associated with a Trojan. These techniques do not offer detection guarantee and might require manual postprocessing diagnosis of suspicious signals or gates.

Proof Carrying Hardware (PCH): A PCH framework uses an interactive theorem prover to verify security properties on soft IP cores. Soft IP cores are synthesizable cores and exist as a netlist or as a hardware description language (HDL) code. A theorem-proving and equivalence-checking approach using PCH to protect IP cores and verify the absence of malicious implants in the IP blocks

Design for Trust

- None of the existing HT detection techniques provide a complete detection guarantee.
- Therefore, embedding HT prevention methods during the design phase through design-for-trust provides a more effective potential approach as a countermeasure against HTs.
- Design-for-trust (DFT) involves the prevention of HT insertion, facilitating the detection of HTs using approaches discussed in the Trojan detection section and trustworthy computing on untrusted components

Design for Trust

Configurable Security Monitors:

- This approach uses security monitors to facilitate real-time functionality monitoring by adding reconfigurable logic in an SoC.

- The signal behavior is checked by feeding the signals to Security Monitors (SMs).

- The configuration of the SM is done to implement FSM, and it does not disturb the normal operation of the system.

- When an abnormality is detected, the checking occurs simultaneously with the other operations of the system and triggers countermeasures as needed.

Design for Trust

Variant-Based Parallel Execution

- This approach performs the simultaneous execution of multiple functionally equivalent variants on different processing elements (PEs), and the results are compared.

- On detecting a mismatch, a new PE is involved until a match is possible and Trojan-infected PEs are identified.

- The more efficient the generation of variants, the more effective is the process. This approach can help computers with multi-core processors achieve a high level of trust, but with an additional cost of computational, performance, and energy overhead.

Design for Trust

Hardware–Software Hybrid Approach:

- Trojan detection and tolerance by a software solution can provide efficient safety for systems with microprocessors.

- This approach uses design-verification tests to identify unused circuitry and marks it suspicious. This technique circumvents HTs by removing the suspicious circuitry and replacing it with exception logic.

- A software exception is triggered which allows the system to bypass the HT and operate normally.

Design for Trust (Prevention of HT Insertion)

Obfuscation:

- This approach is based on making HT insertion difficult for an attacker by obscuring functionality and structural properties of a design.
- The obfuscation approach is based on applying a key-based obfuscation technique.
- This technique modifies a circuit's state transition function, which makes it operate in two modes, i.e., a normal mode and an obfuscated mode.
- An **obfuscated mode** produces incorrect functional behavior, generating incorrect output, while the **normal mode** produces desired output.
- Thus, internal circuit nodes are obscured, and it makes it difficult to identify genuine functionality.
- This approach thwarts the ability of an adversary to insert Trojans without knowing the circuit functionality and right input vectors. Also, a Trojan might be active only in obfuscated mode, which makes it benign.
- The right function of the circuit appears only when the right key is applied. Tamper-proof memory inside the design is used to store the correct key

Design for Trust (Prevention of HT Insertion)

Camouflaging

- It is another obfuscation technique done at the layout level.
- In a camouflaged logic gate, fake connections are added between the layers to create indistinguishable layouts.
- This technique of adding dummy contacts prevents the attackers from obtaining a correct gate-level netlist of a circuit from the layout.
- Thus, hindering the attacker's ability to insert an HT in the original design.

Design for Trust (Prevention of HT Insertion)

Layout Filler

- One of the common ways that attackers use to insert an HT in a circuit is by placing the Trojan in unused vacant spaces in the circuit.
- To prevent HT insertion, the empty spaces in the circuit layout are filled with functional filler cells using a built-in self-authentication (BISA) approach.
- The reason why filler cells are functional is so that the attacker cannot replace these filler cells with a Trojan without changing the functionality of the circuit.
- The functional filler cells form a combinational circuitry.
- Any abnormal function detected during testing indicates the replacement of filler cells with a Trojan

Split Manufacturing

- It is another method to minimize risks to an IC design is the split manufacturing process.
- The objective of this approach is to enhance the security of an IC by hiding the design intent and preventing malicious insertion.
- The design is divided into Front End of Line (FEOL) and Back End of Line (BEOL) parts by different foundries for fabrication.
- The **FEOL layers**, i.e., transistors and low-metal layers, are fabricated in an **untrusted foundry** using an advanced process.
- The **untrusted foundry** then ships wafers to a **trusted foundry**.
- The **trusted foundry** uses a less advanced process to **fabricate back end of line layers**, i.e., high-level interconnects.
- The untrusted foundry cannot find places in a circuit for Trojan insertion as it does not have access to the BEOL layers

Thank you