IIS5008 Hardware Security

Program Assignment 2: Hardware Trojan Insertion

(Due Date: 2025/05/02 (Fri.) 23:59:59)

Background

The modern chip design flow (Fig 1) involves collaborations between various entities, including SoC integrators, 3PIP vendors, and offshore foundries. This complex supply chain introduces trustworthiness issues and creates opportunities for hardware Trojan attacks. Hardware Trojans are malicious additions or modifications to integrated circuits (ICs) that compromise functionality, confidentiality, or performance. They are challenging to detect and remove due to the lack of golden designs and their stealthy behavior.

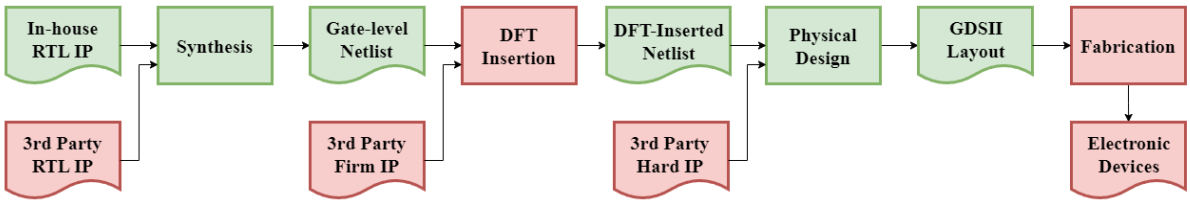

Fig 1. Modern SoC design flow and threat model

Hardware Trojans typically consist of two parts (Fig 2): the trigger and the payload. The trigger monitors internal signals and activates the payload under specific conditions. Trojans can be classified as combinational or sequential based on their trigger mechanism. The payload can have various functions, including altering device behavior, degrading performance, leaking information, or causing denial of service.

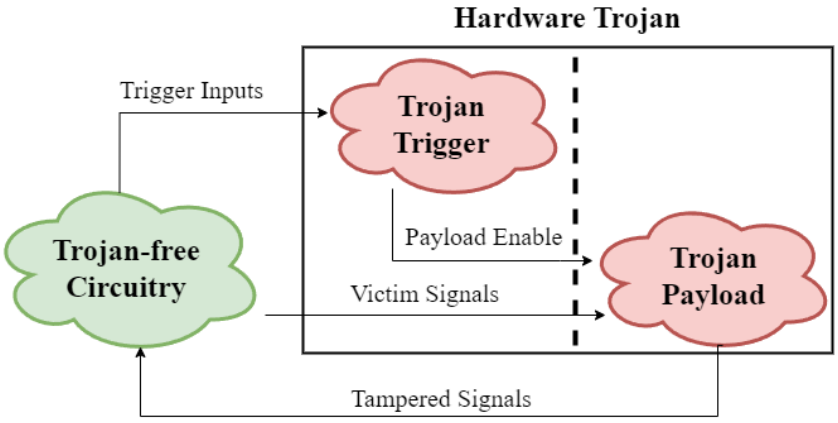

Fig 2. High-level Trojan diagram

Description

The major purpose of this programming assignment is providing opportunities for students to deeply understand hardware Trojan insertion at RTL design level. Therefore, this programming assignment can be divided into two parts: First, you need to complete an AES-128 cryptographic engine at RTL level as a good and honest designer. Second, you need to implant at least one hardware Trojan in this system.

1. Implement an AES-128 cryptographic engine:

In this part, you're tasked with completing the AES-128 system provided to you. AES-128 encryption is a symmetric key encryption algorithm used to protect the confidentiality of data. In AES-128 encryption (Fig 3), data is divided into blocks of 128 bits and encrypted using a 128-bit key. The encryption process involves 10 rounds of **SubBytes, ShiftRows, MixColumns,** and **AddRoundKey,** with each round using the key to transform the data block. The final round does not include the **MixColumns** operation. Eventually, the data block undergoes multiple rounds of operations and is output as encrypted data.

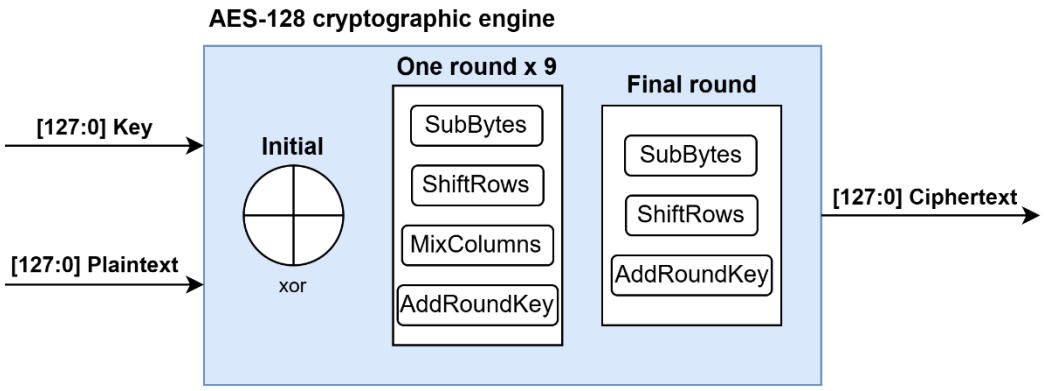

Fig 3. AES-128 cryptographic engine

- **Key and state:**

  The 128 bits of the key and state variable correspond to the entries of the 4x4 array (Fig 4).

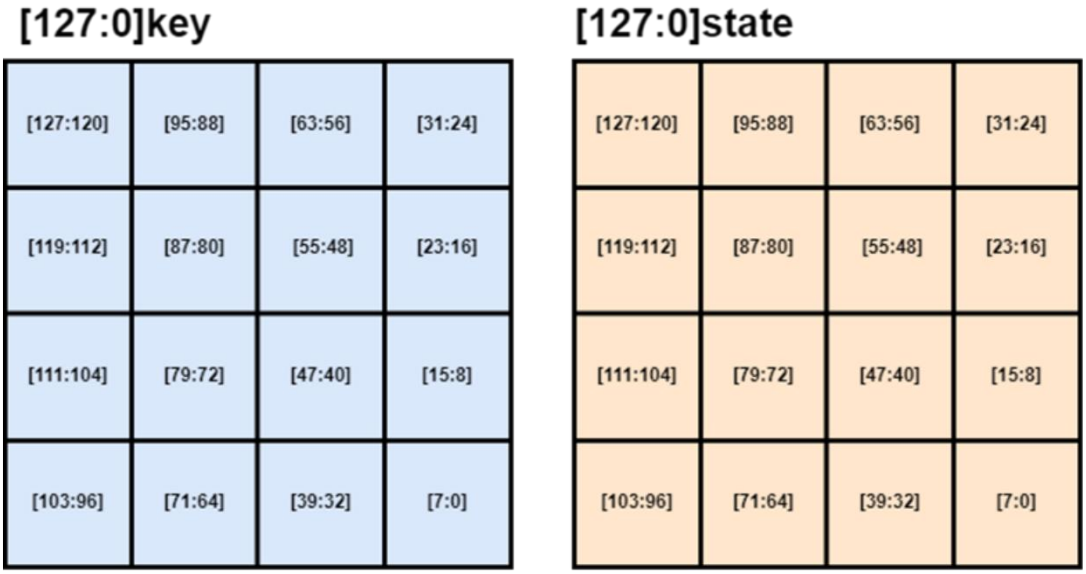

Fig 4. The key and state correspond to the entries of a 4×4 array

- **SubBytes:**

  This step provides the encryption's non-linear transformation capability. It involves substituting each byte of the state with the corresponding byte from the S-box. You need to perform this substitution operation on each byte of the state in every round.

  We provide module S in table.v, which implements the functionality of S-box transformation. Please use this function to complete the SubBytes step.

- **ShiftRows:**

  This step involves cyclically shifting each row of the state to the left. After ShiftRows, each column in the matrix is composed of elements from different rows in the input matrix.

- **MixColumns:**

  In this step, each column's four bytes are combined through a linear transformation. This step operates on the columns of the state by performing matrix multiplication to mix the data.

  We provide module xS in table.v, which simplifies the matrix multiplication for MixColumns. Please use this function to complete the MixColumns step.

- **AddRoundKey:**

  In this step, the round key is XORed with the original matrix. In each encryption iteration, a round key is generated from the main key, matching the size of the original matrix. This round key is then XORed with each corresponding byte in the original matrix.

You've been provided with several Verilog files: "AES_top.v", which acts as the top module. We'll assess the correctness of your code by invoking this top module during testing. "aes_128.v", responsible for implementing the AES-128 process including initial XOR, calling the one round module, calling the final round module, and calling the expand_key_128 module. Additionally, there's the "expand_key_128" module, which handles the key expansion process. The "table.v" file contains essential modules like 8-bit S-box transformation.

- **Complete the AES-128 system:**

Your objective is to complete the implementation of two modules: **"one_round"** and **"final_round"** in the file named as **"round.v"**. These modules are crucial components of the AES-128 encryption process. The "one_round" module is responsible for executing a single round of AES-128 encryption, including SubBytes, ShiftRows, MixColumns, and AddRoundKey operations. On the other hand, the "final_round" module is tasked with executing the final round of AES-128 encryption, which excludes the MixColumns operation.

By completing these two modules, you'll contribute to the overall functionality of the AES-128 encryption system, enabling it to securely encrypt data according to the AES standard.

2. Implement the hardware Trojan you designed in this system:

In this part, you need to implement a hardware Trojan for an AES system, including triggers and payloads.

You can achieve basic scores by implementing a sample hardware Trojan. To earn higher scores, you can re-implement existing hardware Trojan papers. If you design a novel hardware Trojan and implement, you may qualify for a bonus. The trigger can be any triggering condition, such as system boot-up, specific time, or specific input. The payload can involve erroneous output or stealing the key, among other destructive behaviors.

- **Sample hardware Trojan:**

The hardware Trojan will be triggered only when the encryption process reaches 200 clock cycles and the last 16 bits of the input data equal 0xC0DE, and its payload causes the output to leak the key.

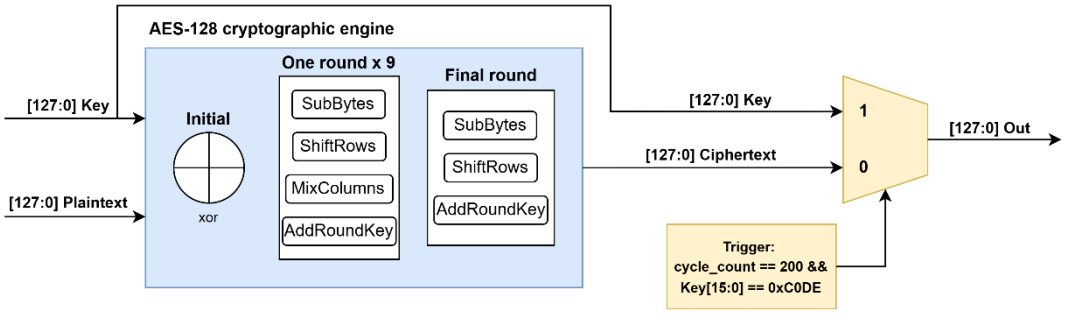

Fig 5. Sample hardware Trojan

- **Hardware Trojan demonstrated in the given paper:**

  There have been many studies on hardware Trojans. This assignment requires you to re-implement a hardware Trojan attack on an AES core, as described in the provided paper: *"A Novel Tampering Attack on AES Cores with Hardware Trojans* [1]*"* by Ayush Jain and Ujjwal Guin.

- **Design a novel hardware Trojan and implement: (bonus)**

  You can also design a novel hardware Trojan by yourself, and you need to explain the Trojan trigger and payload. We will use patterns to test the security of your system, and if the results are correct, you will be awarded points.

Submission Requirement

You have to write this program in Verilog or SystemVerilog and use NC-Verilog as the simulation platform. You MUST implement the assignment independently. If you refer to any external materials, you must cite them properly and demonstrate a thorough understanding of the content.

You must submit three files in this programming assignment. **Please ensure that you adhere to the naming rule mentioned below. Otherwise, your program will not be graded.**

1. StudID_PA2_AES.zip (ex: 9862534_PA2_AES.zip):

   Please compress all your source codes (all .v files) related to the purely AES system in part one (not the entire project) into a single compressed file named StudID_PA2_AES.zip.

2. StudID_PA2_HT.zip (ex: 9862534_PA2_HT.zip):

   Please compress all your source codes (all .v files) related to the hardware Trojan you implemented in part two (not the entire project) into a single compressed file named StudID_PA2_HT.zip.

3. StudID_Name_PA2_report.pdf (ex: 9862534_陳聿廣_PA2_report.pdf):

   A report named StudID_Name_PA2_report.pdf. Note that the only acceptable report file format is .pdf, no .doc/.docx or other files are acceptable.

Report

We don't restrict the report format and length. In your report, you must at least describe:

- How to compile and execute your program; (You can use screenshot to explain)
- The completion of the assignment; (If you complete all requirements, just specify all)
- The hardware Trojan you design; (Including Trojan trigger and payload)
- Your simulation waveform and explain it;

You can also put anything related to the PA in your report, such as pseudocode, control flow diagram, programming developing thought, etc.

Grading

The grading is as follows:

1. Complete the AES-128 cryptographic engine: 30%
2. Sample hardware Trojan implement: 30%
3. Reference hardware Trojan implement: 10%
4. The report: 10%
5. Demo : 20%
6. Design a novel hardware Trojan and implement: 10% (bonus)

Please submit your assignment on time. Otherwise, the penalty rule will apply:

1. Within 24 hrs delay: 20% off
2. Within 48 hrs delay: 40% off
3. More than 2 days: 0 point

Be sure to attend a demo session (the time will be announced later). Otherwise, your program will not be graded.

Reference

[1] A. Jain and U. Guin, "A Novel Tampering Attack on AES Cores with Hardware Trojans," *2020 IEEE International Test Conference in Asia (ITC-Asia)*, Taipei, Taiwan, 2020, pp. 77-82, doi: 10.1109/ITC-Asia51099.2020.00025.