CYENG 225 – SP 23: Microcontroller Essentials for Cyber Applications
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Assignment 4: Chapter 4 - Trusted Execution Environments

Total Points: 100**; and Deadline:** February/23/2023, 11:59 PM.

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and they cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All of your responses to the questions of assignment should be included in a single compressed file to be uploaded to the Gannon University (GU) – Blackboard Learn environment.

Question. Provide short answers (i.e., no more than five lines on average with the font size of 12) for the following items. The grade for each item is **10 points**.

1. Explain the relationships between **TCB** and **TEE**.
2. Mention two main security properties of **TEE** and how they are established.
3. Discuss different methods that are used to provide the security properties of **TEE**.
4. Specify multiple examples of secure processor architectures that provide **TEE**.
5. Mention the assumption and the limitations for provision of **TEE**.

Question 2. Complete the following Steps. The grade for this question is **50 points**.

A. Perform the procedure from the "**EXPERIMENT #1: Introduction to Xilinx's FPGA Vivado HLx Software**" laboratory assignment for the **Advanced Encryption Standard (AES)** algorithm. Samples codes are available in the following for your kind reference.
B. Perform the following tasks:
   - Compile the implementation of the **Advanced Encryption Standard (AES)** algorithm written in the C/C++ programming language as a static binary. Samples codes are available in the following for your kind reference.
   - You are required to the program in the gem5 simulator system and change the CPU model, CPU frequency, and memory configuration and describe the changes in performance.
   - Change the CPU model from **TimingSimpleCPU** to **MinorCPU**. **Hint**: You may want to add a command line parameter to control the CPU model.
   - Vary the CPU clock from **1 GHz** to **3 GHz** (in steps of **500 MHz**) with both CPU models. **Hint**: You may want to add a command line parameter for the frequency.
   - Change the memory configuration from **DDR3_1600_x64** to **DDR3_2133_x64** (DDR3 with a faster clock) and **LPDDR2_S4_1066_x32** (low-power DRAM often found in mobile devices).
C. Execute an attack on the **Advanced Encryption Standard (AES)** algorithm and reperform the **Steps A and B**.
D. Include the following items in your submitting package:
   - Provision of the implementation results from the **Step A**.
   - All files of the implementation of **Advanced Encryption Standard (AES)** algorithm for the **Steps A and B**.

- A file named "**AES-config.py**" (and any other necessary files) that was used to run the gem5 simulations. This file should be set up to use **TimingSimpleCPU** at **1 GHz** and **DDR3_1600_x64** by default.
- Provide a report that includes: (**1**) your overall understanding and conclusions from completing the experiments; (**2**) the interesting points and the challenges that you faced in this laboratory; and (**3**) the screenshots for all of the major steps in your experiments. The report should contain answers to the following questions:
  - ✓ Which CPU model is more sensitive to changing the CPU frequency? Why?
  - ✓ Which CPU model is more sensitive to the memory technology? Why?
  - ✓ Is the AES program more sensitive to the CPU frequency or the memory technology? Why?
  - ✓ If you were to use a different program, do you think your conclusions would change? Why?

Resources for Step A:

1. secworks/aes: Verilog implementation of the symmetric block cipher AES (Advanced Encryption Standard) as specified in NIST FIPS 197. This implementation supports 128 and 256 bit keys. (github.com)
2. mematrix/AES-FPGA: AES加密解密算法的Verilog实现 (github.com)
3. pnvamshi/Hardware-Implementation-of-AES-Verilog: Hardware Implementation of Advanced Encryption Standard Algorithm in Verilog (github.com)
4. ahegazy/aes: Advanced encryption standard implementation in verilog. (github.com)
5. siamumar/tinyAES (github.com)

Resources for Step B:

1. dhuertas/AES: AES algorithm implementation in C (github.com)
2. openluopworld/aes_128: Implementation of AES-128 in pure C. No modes are given. Only one block of encryption and decryption is given here. (github.com)
3. SergeyBel/AES: C++ AES implementation (github.com)
4. ilvn/aes256: A byte-oriented AES-256 implementation. (github.com)
5. polfosol/micro-AES: A minimalist implementation of AES algorithms in C (github.com)
6. exscape/AES: A simple AES implementation in C (with AES-NI support for supported CPUs) (github.com)
7. Yunyung/Cryptography-AES-implement-in-C: Implement AES(Advanced Encryption Standard) Stystem in C program (github.com)
8. kokke/tiny-AES-c: Small portable AES128/192/256 in C (github.com)
9. Source Browser (apple.com)
10. cipher Directory Reference (oryx-embedded.com)
11. aes.c - lib/crypto/aes.c - Linux source code (v6.3-rc2) - Bootlin
12. ceceww/aes: C++ implementation of a 128-bit AES encryption/decryption tool. (github.com)
13. gauss.ececs.uc.edu/Courses/c653/extra/AES/AES_Encrypt.cpp

Resources for Step C:

1. A Survey of Hardware Trojan Taxonomy and Detection | IEEE Journals & Magazine | IEEE Xplore
2. Hardware Trojan: Threats and emerging solutions | IEEE Conference Publication | IEEE Xplore
3. Hardware Trojan Attacks: Threat Analysis and Countermeasures | IEEE Journals & Magazine | IEEE Xplore
4. Experiences in Hardware Trojan design and implementation | IEEE Conference Publication | IEEE Xplore
5. A survey on hardware trojan detection techniques | IEEE Conference Publication | IEEE Xplore

Resources for Step D:

1. ug906-vivado-design-analysis.pdf • Viewer • AMD Adaptive Computing Documentation Portal (xilinx.com)
2. How to Analyse Area, Delay And Power In Xilinx Software ? - YouTube
3. 63 - Vivado's Timing Reports - YouTube
4. Power Estimation and Analysis using Vivado - YouTube
5. Timing analysis with Vivado tools (Part 1) - YouTube
6. Analyzing Implementation Results - YouTube
7. Timing, power and area analyze (xilinx.com)