CYENG 351 – SP 23: Embedded Secure Networking
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Assignment 5: Chapter 5 - Embedded Security

Total Points: 100**; and Deadline:** April/13/2023, 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 1. 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 **20 points**.

1. Mention the type of protocol architectures that requires **Application Programming Interface (API)**. Specify its API rules and explain why they are needed for that type. Provide a figure that reflects your explanations.
2. Specify the **five main rules** in **Design and Implementation of Security Protocols** and explain each briefly.

Question 2. Complete the laboratory part, titled "**Wireshark Lab: TCP v8.1**". The grade for this question is **40 points**. Provide the screenshots for all of the major steps/processes in your experiments as well as the answers to the laboratory questions.

Question 3. Complete the following Steps. The grade for this question is **40 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. Apply statistics to problem solving in Embedded Network Security: In order to analyze the security strength and the encryption performance of your AES implementation, perform certain statistical analyses, including **Histogram Analysis**, **Adjacent Pixel Correlation (APC) Analysis**, **Information Entropy (IE) Analysis**, **Analysis of Number of Pixels Change Rate (NPCR)**, **Analysis of Unified Average Changing Intensity (UACI)**, **Analysis of Encryption Quality (EQ)**, and **Analysis of Mean Absolute Error (MAE)** on the encrypted images (by the AES implementation) as well as the original images. Samples codes are available in the following for your kind reference. Also, refer to the provided papers in the "**AES Security**" compressed package for more information about executing security analyses using statistical methods. You can use MATLAB or Python to implement and perform the statistical analyses.
C. 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**.
   - 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.

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. 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)