Lab 2
Encryption

1 Introduction and Goals

The goal of this lab is to get you familiar with implementing cryptography algorithms. You will create a RSA and a AES implementation in C.

You may only use:

- The default C standard libraries (e.g., `stdio.h`, `stdlib.h`, `string.h`)
- The math library (`math.h`)

This restriction is intentional: the goal is to help you understand how these algorithms work internally rather than relying on pre-built functions.

1.1 Graded Items

For both encryption problems, you will be rewarded extra credit if you go above and beyond the requirements (e.g. supporting any size plaintext, particular fast implementations, adding integrity checking, etc.)

1. Problem 1: Screenshot of the output of the AES test program.
2. Problem 2: Screenshot of the output of the RSA test program.
3. Include all code files, test input files (if you have them), etc. in the final zipped submission.
4. Answer all of the questions asked for each problem in your report as well.

Lab reports must be in readable English and not raw dumps of log-files.

2 Problem 1: AES CTR Mode

In this problem, you will write a software algorithm that implements AES with the CTR block cipher mode. We will provide you with the S-box, and you will be responsible for implementing encrypting and decrypting a 256-bit message.
The `part1/aes.c` file provided has the starter code that provides you with the generated key and IV.

1. Submit your screenshot of your AES implementation running as **Figure 1** in your report.

2. Describe how you would integrate integrity checking on top of your AES implementation in your report.

3 Problem 2: RSA

In this problem, you will write a software algorithm that implements RSA. We will provide you with a generated 128-bit key, and you will be responsible for implementing encrypting and decrypting of 128-bit plaintext messages.
The `part2/rsa.c` file provided has the starter code that provides you with the modulus, public exponent, and private exponent of the generated key.

1. Submit your RSA implementation screenshot as **Figure 2** in your report.

2. Describe how you would integrate integrity checking on top of your RSA implementation in your report.

Your lab reports must be typed and must not exceed 6 pages. You are encouraged to use the report template provided on Canvas. Please submit your lab report and all of your code in a zip/tar file on on Canvas as `lab2_EID.tar.gz` or `lab2_EID.zip`. Include both files as well as any necessary `.h` files in your final zip file.