ECE 471 Lab 4

RSA Public-Key Encryption and Signature Lab

Mitchell Dzurick

3/23/2020

Github with all documentation - `https://www.github.com/mitchdz/ECE471`

Contents

RSA Public-Key Encryption and Signature Lab

 The development of this document was partially funded by the National Science Foundation under Award No. 1303306 and 1718086.

Overview

RSA (RivestShamirAdleman) is one of the first public-key cryptosystems and is widely used for secure communication. The RSA algorithm first generates two large random prime numbers, and then use them to generate public and private key pairs, which can be used to do encryption, decryption, digital signature generation, and digital signature verification. The RSA algorithm is built upon number theories, and it can

be quite easily implemented with the support of libraries. The learning objective of this lab is for students to gain hands-on experiences on the RSA algorithm. From lectures, students should have learned the theoretic part of the RSA algorithm, so they know mathematically how to generate public/private keys and how to perform encryption/decryption and signature generation/verification. This lab enhances student's understanding of RSA by requiring them to go through every essential step of the RSA algorithm on actual numbers, so they can apply the theories learned from the class. Essentially, students will be implementing the RSA algorithm using the C program language. The lab covers the following security-related topics:

- Public-key cryptography
- The RSA algorithm and key generation
- Big number calculation
- Encryption and Decryption using RSA
- Digital signature
- X.509 certificate

Lab Environment. This lab has been tested on our pre-built Ubuntu 12.04 VM and Ubuntu 16.04 VM, both of which can be downloaded from the SEED website.

1 Task 1: Deriving the Private Key

1.1 Task 1: Solution

2 Task 2: Encrypting a Message

2.1 Task 2: Solution

3 Task 3: Decrypting a Message

3.1 Task 3: Solution

4 Task 4: Signing a Message

4.1 Task 4: Solution

5 Task 5: Verifying a Signature

5.1 Task 5: Solution

6 Task 6: Manually Verifying an X.509 Certificate

6.1 Task 6: Solution