Programming Project: Crypto Systems (20 pts)

Single-person teams choose two from the following three problems. Other teams need to complete all three problems.

Problem 1. Modern Symmetric Cipher & Decipher Implementation

Program to implement the DES or AES algorithm. You can choose either one based on your interest and use any programming language you like, such as C, C++, Java, and python. Directly using des or aes libraries receives 0 for this problem. Your program does not need to encode a plaintext exactly the same as the existing library function. But you need to specify in your report how you implement each component of the cipher/decipher in both text and code segments. And how many rounds? You can refer to the DES or AES algorithm designs in the lecture notes or some similar designs. Show that your encryption and decryption function work using example plaintexts.

Problem 2. RSA Crypto System. 1) Write a prime number check program to check any input number to be whether a prime number. 2) Find the 10th and the 19th prime numbers p and q between 1000 and 10000 to build an RSA crypto system. Write down the public key PU = {e, n} and the private key PR = {d, p, q}. 3) Program to implement the encipher and decipher. Test your RSA crypto system by encrypting and decrypting a message “rsa” (Map each letter to 0 - 25). 4) If an adversary obtains the public key PU = {e, n}, demonstrate how the adversary uses the exhaustive search to get the private key d and show the time cost of the search.

Problem 3. SHA-256. In this project, you will program to examine the complexity of a crypto-based puzzle, which is used as the foundation of cryptocurrency. You are allowed to use whatever library you desire. Of course, some languages will have advantages over other languages.

Your task is to implement a hash-based crypto puzzle using SHA-256. The way the crypto puzzle works is as follows. Alice issues a challenge to Bob: Alice creates a puzzle P that is *B*-bit long, where *B* < 256. She challenges Bob to find a message *M* that, when fed into the SHA-256 function *h()* yields the last *B* bits equal to P. That is, $h(M) = [***\cdots***,P]$. Bob attempts to find such an M by generating random guesses and hashing them. When he finds one, he reports it to Alice and has solved the crypto puzzle. You are to estimate the amount of time it takes to solve the hash-based crypto puzzle for different values of *B* (for B =4, 8, 12 and 16). Plot a figure to show that larger B leads to exponentially high time costs to break the puzzle. Sometimes you could be lucky and guess M on your first try. Therefore, you will need to average over an amount of trials in order to get an accurate time estimate. You will need to describe the approach you used and the explain your observations. Make certain to describe which language and libraries you used (if they are public).

===

You are required to write a project report by solving the above problems. Describe your design clearly and your observations. Submit a copy of your code. Your grade will be based upon the clarity and thoroughness of your report.