Fall 2019: CSCI 430: Introduction to Computer and Network Security

Assignment 2

Deadline. Oct 9, 2019, 9pm.

Programming language. The preferred programming language is Java, but you could use any language. If we cannot compile your program, you would need to come to office hours and demonstrate your code on your computer.

Submission instructions. Create a private git repo on bitbucket and share the repo with the instructor (mnaveed@usc.edu) and TA (hsaleem@usc.edu). For full credit, only the code pushed to bitbucket before the deadline will be considered. Your programs need to compile without any additional software and produce required output. In addition to the code, you will need to submit nonce on Piazza for problem 4.

Problem 1: Write two programs, one for public key encryption and another for hybrid encryption, to compare the performance of public key encryption and hybrid encryption. Your programs should have a sender function and a receiver function, and you could use the main program as a communication medium to transfer the data between the sender and the receiver. Your programs should take as an input a path to a large file stored on the disk, encrypt it, and then decrypt the resulting ciphertext, and output the encryption and decryption time. It is important that the input file be large because the difference in performance may not be noticeable with small files.

Problem 2: Write a program that takes as input from the user a path to an arbitrary file stored on disk and outputs a digital signature on this file and the verification key. Your program should also allow the user to enter the file path, the digital signature, and the verification key, and outputs “valid” if the signature is valid and “invalid” otherwise.

Problem 3: Write a program for an append-only ledger. Your program should take fixed length strings as an input from the user, append it to the ledger, and store this ledger on hard disk. When you execute the program, it should load the previously stored ledger from the hard disk, append any new strings, if any, to the ledger and store the new ledger to the hard drive again. Every time, the ledger is loaded from the hard drive, your program should ensure that it has not been tampered.

Problem 4: Write a program to find a nonce such that SHA256(nonce || ***your-name***) produces a hash that starts with at least 9 zeros in hexadecimal or 36 zeros in binary. **your-name** is your full name, e.g., if I were to do this problem, it would be SHA256(nonce || muhammadnaveed), and || denotes concatenation. In addition to submitting your code, there will a thread on Piazza to post your nonce and you must post your nonce to that thread before the deadline to get credit for this problem. **Your program may take more than a week to find such a nonce, so please start early.**