**Fall 2023 CS120 Homework Assignment #4**

**Due Date: Jan. 7, 2024**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Please read the following instructions carefully before answering the questions:**

* This assignment is to be completed by each student **individually**.
* There are a total of **7** questions.
* When you write your answers, please try to be precise and concise.
* Fill your name and student ID at the first page.
* Please typeset the file name and format of your submission to the following one: YourID\_CS120\_HW4.pdf (Replace “YourID” with your student ID). Submissions with wrong file name or format will **NOT** be graded.
* Submit your homework through Blackboard.

1. (10 points) DNS

(a) Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department? Explain.

(b) Suppose that your department has a local DNS server for all computers in the department. You are an ordinary user (i.e., not a network/system administrator). Can you determine if an external Web site was likely accessed from a computer in your department a couple of seconds ago? Explain.

1. (30 points) In this problem, we explore the Diffie-Hellman (DH) public-key encryption algorithm, which allows two entities to agree on a shared key. The DH algorithm makes use of a large prime number p and another large number g less than p. Both p and g are made public (so that an attacker would know them). In DH, Alice and Bob each independently choose secret keys, SA and SB, respectively. Alice then computes her public key, TA, by taking g^SA mod p. Bob similarly computes his own public key TB = g^SB mod p. Alice and Bob then exchange their public keys over the Internet. Alice then calculates the shared secret key S by TB^SA mod p. Similarly, Bob calculates the shared key S′ by TA^SB mod p.

(a) Prove that, in general, Alice and Bob obtain the same symmetric key, that is, prove S=S’.

(b) With p = 11 and g = 2, suppose Alice and Bob choose private keys SA = 5 and SB = 12, respectively. Calculate Alice’s and Bob’s public keys, TA and TB.

(c) Provide a timing diagram that shows how Diffie-Hellman can be attacked by a man-in-the-middle. The timing diagram should have three vertical lines, one for Alice, one for Bob, and one for the attacker Trudy.

1. (10 points) Suppose a file contains the letters *a*, *b*, *c*, and *d*. Nominally we require 2 bits per letter to store such a file.
2. Assume the letter *a* occurs 50% of the time, *b* occurs 30% of the time, and *c* and *d* each occurs 10% of the time. Give an encoding of each letter as a bit string that provides optimal compression. (Hint: Use a single bit for *a*.)
3. What is the percentage of compression you achieve above? (Compared with using 2 bits per letter)
4. (10 points) How might you encode audio (or video) data in two packets so that if one packet is lost, then the resolution is simply reduced to what would be expected with half the bandwidth? Explain why this is much more difficult if a JPEG-type encoding is used.
5. (20 points) Consider the following simplified BitTorrent scenario. There is a swarm of peers and, during the time in question, no peers join or leave the swarm. It takes a peer 1 unit of time to upload or download a piece, during which time it can only do one or the other. Initially, one peer has the whole file and the others have nothing.
6. If the swarm’s target file consists of only 1 piece, what is the minimum time necessary for all the peers to obtain the file? Ignore all but upload/download time.
7. Let be your answer to the preceding question. If the swarm’s target file instead consisted of 2 pieces, would it be possible for all the peers to obtain the file in less than time units? Why or why not?
8. (10 points) One mechanism for resisting replay attacks in password authentication is to use *one-time passwords*: A list of passwords is prepared, and oncehas been accepted the server decrements and prompts for next time. At a new list is needed. Outline a mechanism by which the user and server need only remember one master password *mp* and have available locally a way to compute . Hint: Let be an appropriate one-way function (e.g., MD5) and let applied times to *mp*. Explain why knowing doesn’t help reveal .
9. (10 points) Suppose we have a very short secret s (e.g., a single bit or even a Social Security number), and we wish to send someone else a message now that will not reveal but that can be used later to verify that we did know . Explain why or with RSA encryption would not be secure choices, and suggest a better choice.