

## **QR 48/CSCI E-2 Problem Set 3**

Due Monday, April 6 at the beginning of class for QR48 and due Tuesday, April 7 at 12:01 AM EDT for CSCI E-2. Show how you got your answers. Use reasonable units (e.g., say “1 second” instead of “1,000,000 microseconds”).

1. Here we will have you brush up on some modular math:
  - a) (1 point) What is  $577 \bmod 13$ ?
  - b) (1 point) Compute  $10^{100}$  (a googol)  $\bmod 10$ . Show your steps.
  - c) (2 points) That may have been easy, what about  $79^{65536} \bmod 101$ ? Again, show your steps.
  
2. Alice and Bob need to communicate secretly. They decide to use a modified version of the Diffie-Hellman protocol that works as follows:
  - As with ordinary Diffie-Hellman, everyone begins with a large prime  $p$  and an integer  $q$  less than  $p$ . Everybody knows these values.
  - When Alice and Bob want to communicate, each chooses a random number as a secret key. Let's call those numbers  $a$  and  $b$ .
  - Alice computes her public key as follows:  $A = (a + q) \pmod p$
  - Bob computes his public key as follows:  $B = (b + q) \pmod p$
  - Alice computes a number by adding her private key to Bob's public key  $\pmod p$ , and Bob does likewise.
  - a) (3 points) Have Alice and Bob actually computed a shared key? Explain.
  - b) (3 points) Is their method secure against eavesdroppers? Explain.
  - c) (2 points) Why might Alice and Bob have chosen this method instead of the regular Diffie-Hellman protocol? What is the distinguishing characteristic of regular Diffie-Hellman that provides its security?
  
3. Digital audio files average around 4MByte/song. Suppose you had a song that you wanted to transmit to your friend (and that you had the legal right to do so). Assume the following:
  - There are 10 hops in the Internet path between you and your friend
  - During each hop, the data can go at a rate of 2.5 megabits/second ( $2.5 * 2^{20}$  bits/second)
  - For each hop, there's a bit error rate of 1 in  $10^7$ , which means that each time a bit is sent along a hop, there's a 1 in  $10^7$  chance that it will be wrong.

Whenever we talk about transmission, we are referring just to the amount of time each hop takes due to the 2.5mbit/sec data rate. (We ignore any delays due to queuing and propagation.)

- a) (1 point) How long would it take to send the song if the entire thing could be sent not in packets, but as a single chunk?
- b) (2 points) What is the probability that the entire song will make it through all 10 hops to the destination without any errors?

Now suppose that we break up the message into packets. Assume that each packet will take an additional 128 bytes of overhead for routing and reassembly data.

- c) (1 point) How long would it take to send the song if it were broken into 512-byte packets (512 bytes of data + 128 bytes of overhead for each packet sent)? Recall that the entire packet must arrive before transmission of the next packet can begin.
- d) (1 point) What about packets with 2 kilobytes of data?
- e) (3 points) What's the probability that an individual packet makes it all the way to the destination without any errors? Include the bytes of overhead in the packet size, and answer for both packet sizes.
- f) (1 point) What conclusions can you draw about the relationships between packet sizes, transmission times, and error probabilities?

We've looked at error rates, but we can't actually detect errors in this scheme.

- g) (1 point) What if we added a parity bit to each byte? Namely, for each byte, we add a 0 or a 1 at the end of it so that each set of 9 bits now has an even number of 1s. How big will our song be now?
- h) (2 points) What's the probability of a 1-bit error on a single byte on a single hop? Would this be detected by our parity code?
- i) (2 points) What's the probability of a 2-bit error on a single byte on a single hop? Would this be detected by our parity code?

This error rate still may be too high for comfort. Suppose, instead, that we use a 64-byte cyclic-redundancy check (CRC) on each packet, which gives an extremely low (low enough to ignore) probability of an undetected error. Now our packets are `data_size + 128` bytes of overhead + 64 bytes of CRC.

- j) (3 points) How long would it take to transmit the song with the 64 byte CRC added to each packet to your friend? Calculate this for both 512-byte packets and 2-kilobyte packets.
4. (1 point) Explain why our friend might or might not care whether the song had been transmitted without error. That is, explain what we would have to know about our friend's intentions to determine whether it was worth it to include error detection.
  5. This problem, worth 6 points total, is for CSCI E-2 graduate students. Other students may do this problem if they'd like for 0 points.

The Md5 hash function is a method of verifying the integrity of data. Suppose you have a video that you wish to transmit to a court to be used as evidence. The court wishes to verify that the video is unchanged when you transmit it to them electronically. The video is 20 Megabytes. Taking an Md5 hash outputs a number that is 32 hexadecimal digits long (128 bits). You can assume for this problem that Md5 hashes are distributed uniformly across the 128 bit hash space.

- a) (1 point) What is the probability that two files of equal length will have the same Md5 hash?
- b) (2 points) What is the expected number of one bit errors that will produce a hash collision, that is, how many movie files that differ by only one bit from the original file would you expect to produce the same hash as the original file?
- c) (3 points) What is the expected number of two bit errors in the movie that will produce a hash collision?