INST346 First Midterm Exam Answers Spring 2018
Corrected on 2/28/2018

1. [5 points] Consider the case of a computer with a one terabyte (TB) hard disk drive with 512 byte sectors that has a 10 millisecond average access time (which means that is able to transfer a different 512 byte sector from disk into main memory every 10 milliseconds). If there is a single 100 megabyte file in which the sectors are placed at random locations on the disk, how long would it take to read that file into RAM (i.e., into the computer's main memory)?

A 100 MB file is 100 times 1024 times 1024 = 104,857,600 bytes (100,000,000 bytes was also an acceptable value to use). With 512 byte sectors, the 100 MB file would require 104,757,600 divided by 512 = 204,800 sectors for storage. If those sectors are distributed randomly on the disk, reading each of these sectors one at a time will require 204,800 times 10 = 2,048,000 miliseconds = 2,048 seconds = 34.1333 minutes. If instead you used 100,000,000 bytes, the same series of calculations would lead to 1953.125 seconds = 32.55 minutes, which was also an acceptable answer.

2. [5 points] Answer BOTH parts of this question:
    a. [3 points] Compute the number of bytes necessary to represent a photograph that is stored as a 1024 dot wide by 768 dot high image file in which each dot is represented as one of 256 values each for the colors red, green, and blue.

256 possible values can be stored in 1 byte (because 2 to the 8th power is 256). Thus storing three values for each pixel (the proper name for a dot; short for picture element) would require three bytes. The picture is made up of 1,024 times 768 = 786,432 pixels. Because each pixel requires 3 bytes, a total of 768,432 times 3 = 2,349,396 bytes will be required. Dividing twice by 1024 to get to Megabytes results in 2.25 MB (although 2.35 MB was also an acceptable answer).

    b. [2 points] If a good lossy compression scheme such as JPEG (i.e., .jpg) is used to compress the resulting file (while maintaining a fairly good picture quality), approximately how many bytes of disk storage would you expect the compressed picture to require? Your answer to this question does not need to be exact – a rough estimate is all that is needed. But your rough estimate should be realistic!

According to Wikipedia, 10:1 compression is commonly achieved by JPEG without significant reductions in the picture quality. This would result in a file that required only one-tenth the storage as the uncompressed file, which in this case would be 225kB (=2.25 MB divided by 10). Other sources on the Web indicate that 20:1 compression is sometimes achieved by JPEG with acceptable picture quality, and of course for some pictures compression somewhat less that 10:1 might be achieved (depending on the amount of fine-grained detail in the picture). Because we asked only for approximate values, we accepted answers between approximately 4:1 (which is considerably less

compression than one would typically expect) to 25:1 (somewhat greater than the highest quoted values we could find reported). Several students answered with considerably lower compression values than would be expected. One cause of these problems might have been the incorrect use of an already compressed reference as a starting point (e.g., asking how much additional compression could be achieved before degradation becomes noticeable).

3. Answer BOTH parts of this question:
    a. [3 points] Describe a two-factor authentication system that is designed to reduce the probability that an unauthorized person would be able to gain access to a computer account.

A good answer to this question will address that two-factor authentication requires two different types of information (selected from something about you, something you have, or something you know). Specific examples for both factors must be provided. For example, one could be a username/password combination, and the other could be an access card that plugs into the computer. If it would be clear to us that your two-factor authentication system would reduce the probability of unauthorized access (compared to the use of either factor alone), simply presenting the design would suffice. If, however, we did not see how the approach you propose would reduce the probability of unauthorized access, we would need to see a convincing explanation of that from you.

    b. [2 points] Describe a realistic situation in which an unauthorized person would be able to gain access to the computer account despite the use of the specific two-factor authentication system that you described in part a.

One way a person could gain access despite the use of two-factor authentication is breaking both factors. For example, an unauthorized person could break into a person's residence, and steal their access card while also taking a photo of a password sheet next to their computer. The unauthorized person could plug the access card into their computer, and enter the password they obtained from the password sheet to sign in, pretending to be the authorized user. A good answer should state that both factors must be broken, and how each of the two factors could be broken.

A second way to gain access despite the use of two-factor authentication is to compromise the machine that manages the two-factor authentication. For example, an unauthorized person could break into one of Google's server facilities and then use their physical access to reboot the machine with a hacked operating system that allowed them to change the two factors for a specific user to anything they like.

Most incorrect answers for this question left out explanation for breaking one of the two factors. Usually this was a case of leaving out how a user's username and password were obtained, or how that information was used to log in as the user.

4. [5 points] Explain why public key encryption (rather than symmetric key encryption) is used to implement digital signatures.

With "public key" encryption it is possible for the sender to encrypt something using a private key and then for the recipient to decrypt it with the corresponding public key. This can be used to create a digital signature by first hashing the message (which is a form of lossy compression) and then encrypting the resulting hashed value using the private key. Anyone who possesses the corresponding public key can then hash the message themselves and compare it to the results that they get form using the public key to decrypt the encrypted hash value. If the two values match, they can be sure that the message was sent by the user who has the corresponding private key. Writing all of that would have answered the question, but shorter answers were also possible if the answer clearly conveyed an understanding of the salient details of that longer answer. One way to have answered more succinctly would have been to simply point out that in a "public key" encryption system, knowing the public key does not reveal the private key, so the message can be encrypted using the private key and anyone who can decrypt the message can be sure of who sent it. That answer omits some details (such as the hash), but it answers the question that was asked (about why public key encryption is used). Another way of answering the question would have been to point out that in a symmetric key system both the sender and every authorized recipient must have the same key, so any recipient would have been able to forge a digital signature from a sender other than themselves if symmetric key encryption had been used. There were several incorrect answers to this question, most of which were of one of two types. One type of incorrect answer explained how public key encryption can be used to achieve secrecy rather than how it can be used for digital signatures. Another type of incorrect answer described public key encryption in general terms and then simply asserted that it could be used for digital signatures without supporting that assertion in any of the three ways described above.

5. [5 points] Both RAID-5 and tape backup can help to avoid loss of data in the event of a hard disk drive failure. Explain ONE reason why RAID-5 is better than tape backup. Then explain ONE reason why tape backup is better than RAID-5.

A good answer to this question will state one reason in favor of tape, and another reason in favor of RAID-5. Some examples are provided above.

Some reasons why RAID-5 could be considered better than tape backup:

- *RAID-5 can store data up until the time of a disk failure, so even the most recent data can be recovered*
- *RAID-5 has faster write times due to stripping*

Some reasons why backup tape could be considered better than RAID-5:

- *Less expensive per GB of data*
- *More portable*
- *Easier to decorrelate risk (e.g., tape can be stored in a different location form the disk(s))*
- *Tape is less susceptible to errors over long periods of time (years).*

Most points lost on this question resulted from stating facts that were true for both tape and RAID-5.

6. [5 points] Explain how specific components of a smartphone could be used by an app to determine where the phone is located (in the world) and how it is oriented (e.g., laid flat on a table with the top pointed east, or held vertically in your hand with the back facing northwest). Then explain which smartphone components could be used by an app to detect changes in the orientation of the phone.

Components used to determine location in the world (position):

- *GPS*
- *WiFi*
- *Cellular network*

Components used to determine orientation (direction):

- *Magnetometer (also referred to as compass) for horizontal direction*
- *Accelerometer (which can detect the force of gravity) to identify vertical direction*

Components used to determine changes in orientation (angular velocity/rotation)

- *Gyroscope*

Most incorrect answers for this question paired sensors with the wrong function (e.g., Accelerometer with location in the world).