EECS 122: Introduction to Communication Networks

Homework 10

(6 points)

Due: 1999-Nov-19-Fri (in class, or 467 Cory by 2pm)

Problem 1. (2 points) Let's define a high-fidelity audio system as one that reproduces frequencies in the range 20 Hz to 20 kHz with a signal-to-noise ratio of 84 dB. Can a physical link with a signal-to-noise ratio of only 7 dB be used to transmit high-fidelity audio? If so, under what conditions?

Problem 2. (2 points) A brute-force attack on an encrypted message simply tries decrypting with every key until one yields an intelligible message. Suppose that today the largest key for which a brute-force attack is feasible is 56 bits. Processor speed has been doubling about every 18 months for the past 15 years or so (this is related to Moore's Law, which states that the number of transistors on a chip doubles every 18 months, which has been true for the past 30 years or so). If this trend continues, and today you encrypt a message using a 128-bit key, about how long can you expect it to remain a secret?

Problem 3.

a) (2 points) DES maps a 64-bit input block and a 56-bit key to a 64-bit output block. Given two 64-bit blocks X and Y, how many keys K are there, on average, such that $\text{DES}(X,K) = Y$ (that is, Y is the encrypted version of X using key K)? (Note that the answer need not be an integer.)

b) (food-for-thought) Why might the answer to part (a) be of any interest?

Problem 4. (food-for-thought) An early version of the Secure Socket Layer (SSL) used the following method for *A* to prove its identity to *B*:

> *A* generates symmetric session key *K*
> *A* sends *K* encrypted with *B*'s public key
> *B* generates nonce *N*
> *B* sends *N* encrypted with *K*
> *A* signs *N* with *A*'s private key, sends result encrypted with *K*

This protocol has an error—at the end, it is possible that *B* will be fooled into thinking it is talking to *A*, when actually it is not. Can you see how? Hint: The protocol can be fixed by having *A* send not only the signed *N* but also the name of *B*, both encrypted with *K*, in the last step. This problem was taken from the paper *Prudent Engineering Practice for Cryptographic Protocols* by Abadi and Needham (obtainable from the Interesting Links section of the class web page).

Problem 5. (hand-on) Use `pgp` to create a public/private key pair for yourself, and have a friend do the same. Send each other email that is private and authenticated.