CS 127/CSCI E-127: Introduction to Cryptography

Problem Set 1

Assigned: Sep. 6, 2013 Due: Sep. 13, 2013 (5:00 PM)

Justify all of your answers. See the syllabus for collaboration and lateness policies. Submit solutions by email to `mbun@seas` (and please put the string "CS127PS1" somewhere in your subject line).

Problem 1. (Expectations)

a) Let X be a random variable that takes non-negative integer values. Prove that $\mathrm{E}[X] = \sum_{i=1}^{\infty} \Pr[X \geq i]$. (Hint: define $\{0,1\}$-valued random variables X_i, where $X_i = 1$ iff $X \geq i$.)

b) Suppose we have a random experiment that "succeeds" with probability p, and we repeat independent trials of the experiment until we obtain the first success. Show that the expected number of trials is $1/p$.

Problem 2. (Arbitrary Random Choices from Coin Flips) Often we describe randomized algorithms as making random choices from arbitrary sets, but sometimes it will be convenient to assume that we only make use of fair coin tosses (i.e. random bits).

Consider the following methods for generating a random number in the interval $\{0, \ldots, N-1\}$. In each, we let $n = \lfloor \log_2(N-1) \rfloor + 1$ be the bit-length of $N-1$ and let $b_{n-1}b_{n-2}\cdots b_0$ be the binary representation of $N-1$ (so $b_{n-1} = 1$).

1. Use n coin tosses to generate a random number M between 0 and $2^n - 1$. If $M < N$, output M. Otherwise repeat.

2. For $i = n-1$ down to 0, do the following:

   - If $b_i = 1$ or there is a $j > i$ such that $c_j < b_j$, then use a coin toss to generate $c_i \xleftarrow{R} \{0,1\}$.
   - Otherwise set $c_i = 0$.

   Output $c_{n-1}c_{n-2}\cdots c_0$ (interpreted as a binary number).

3. Use $n+10$ coin tosses to generate a random number M between 0 and $2^{n+10} - 1$. If $M < N \cdot \lfloor 2^{n+10}/N \rfloor$, output ($M$ mod N). Otherwise, repeat.

For each of the above methods, (a) say whether its output is uniformly distributed in $\{0, \ldots, N-1\}$, and (b) compute the expected number of coin tosses used. Which method would you prefer if N is a "typical" 128-bit number? Why?

(Extra Credit!) Show that for every k that there is no procedure that samples uniformly from $\{0,1,2\}$ while *always* using at most k coin flips.

Problem 3. (Vigenère Cryptanalysis) 19th century writer and mathematician Charles L. Dodgson (Lewis Carroll) famously asserted that the Vigenère cipher was "unbreakable." Ironically, it was some of his contemporaries who devised generic methods of cracking the cipher. Just how confident should Carroll have felt about passing around, say, encrypted preprints of *Alice in Wonderland*?

a) Write a fully automated program for breaking **shift ciphers** of plaintexts written in English. The table of frequencies of English words from page 13 of Katz-Lindell (1st ed.) has been reproduced in the provided file `english_frequencies.txt` for your convenience. You may use the strategy suggested on pages 13–14 of Katz-Lindell (1st ed.) and/or your own ideas. In particular, we encourage you to experiment with different measures of distance between letter distributions.

   In prose, describe what your program does and what the main ideas are, and justify your major design choices. Attach the source code in an appendix to your submission.

b) Run your program on the challenge ciphertexts contained in `shift_ciphers.txt`. It's ok if your program can't decrypt all of them on its own. Submit the output of your program.

c) Building on your program for shift ciphers, write a program for breaking the **Vigenère cipher** for known key length. (As an extra challenge, you can try to decrypt under unknown key lengths.)

   In `alice.txt`, we have provided you with the text of *Alice in Wonderland*, divided into blocks of 20 characters. Pick a random Vigenère key k of length 3. Find the minimum number of blocks that, when encrypted under k, can be successfully decrypted by your program. Repeat this exercise for some different key lengths ranging from 1 to 1000, and draw a plot of key length against the number of blocks required.

   The full text of *Alice in Wonderland* is 107,720 characters long (without spaces or punctuation). About how long an encryption key would Carroll have needed to secure his book from your program?