CS 120/CSCI E-177: Introduction to Cryptography

Problem Set 1

Assigned: Sep. 28, 2006 Due: Oct. 4, 2006 (1:10 PM)

Justify all of your answers. See the syllabus for collaboration and lateness policies. You can submit by email to `ciocan@eecs` (please include source files) or by hardcopy in the CS 120 box in the basement of Maxwell Dworkin.

Problem 1. (Expectations)

1. 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 $X \geq i$.)

2. 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 $n = \lceil \log_2 N \rceil$ be the bit-length of N and let $b_{n-1}b_{n-2}\cdots b_0$ be the binary representation of N (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?

Problem 3. (More examples of perfect secrecy)

1. Prove that the substitution cipher for messages of length 1 satisfies the definition of perfect secrecy.

2. Prove that the shift cipher for messages of length 1 satisfies the definition of perfect secrecy.

3. Suppose Alice & Bob wish to encrypt a 1000-bit message with perfect secrecy. Would you recommend they use the one-time pad, the substitution cipher over an alphabet of size 2^{1000}, or the shift cipher an alphabet of size 2^{1000}? Compare the advantages and disadvantages of the three possibilities.

Problem 4. (Encrypting Random Data) In class, it was asked if it is easier to achieve security if we know that we are encrypting 'random data' rather than English text. In this problem, you will see an example illustrating why we need to be very careful even in this setting.

Because of security concerns, the system administrator Sid Sysop of the Megaware Corporation suggested that certain confidential communications from the CEO to the employees be encrypted using the one-time pad encryption scheme (G, E, D). The problem, of course, is that the one-time pad can only be used once. So the key for the one-time pad is replaced every hour. To do the key refreshing, the new key $k_t \xleftarrow{R} G$ at a given time t is sent to each employee e encrypted using a long-term key $k_e \xleftarrow{R} G$ held by the employee.[1] For example, on a given 8-hour workday, Sid generates independent one-time pads $k_1, \ldots, k_8 \xleftarrow{R} G$, and over the course of the day, sends employee Alice the ciphertexts $E_{k_A}(k_1), \ldots, E_{k_A}(k_8)$, where E is the one-time pad encryption algorithm. Over the course of the day, the CEO might send up to 8 messages $m_1, \ldots, m_8$, encrypted as $E_{k_1}(m_1), \ldots, E_{k_8}(m_8)$. Given her key k_A, Alice can decrypt the ciphertexts from Sid to obtain the keys $k_1, \ldots, k_8$, which then enable her to decrypt the ciphertexts from the CEO and obtain the messages $m_1, \ldots, m_8$.

Notice that the key k_A is used multiple times, contrary to the usual warnings about the one-time pad. But, reasons Sid Sysop, it is only used to encrypt messages k_t that are chosen *uniformly at random.* The one-time pad has the property that $E_{k_A}(k_t) = k_t \oplus k_A = E_{k_t}(k_A)$, so by the perfect secrecy of the one-time pad, this ciphertext reveals no information about k_A when k_t is random. Thus, k_A should remain a 'good key' and be safe to reuse for encrypting k_{t+1}.

Show that, despite Sid's intuition, this system is actually insecure. Specifically, show how one can gain potentially useful information about the messages $m_1, \ldots, m_8$ from observing the ciphertexts $E_{k_A}(k_1), \ldots, E_{k_A}(k_8), E_{k_1}(m_1), \ldots, E_{k_8}(m_8)$ being sent.

[1] Each employee has a different key to ensure that they cannot continue to read the communications after they leave the company.