

## Problem Set 1

This problem set is due *via email*, to `6857-hw@mit.edu` on *Wednesday, February 20* by the beginning of class.

You are to work on this problem set in groups of three or four people. Problems turned in by individuals, pairs, pentuples, etc. will not be accepted. Be sure that all group members can explain the solutions. See Handout 1 (*Course Information*) for our policy on collaboration. If you do not have a group, let us know.

*Homework must be submitted electronically!* Each problem answer must appear on a separate page. Mark the top of each page with your group member names, the course number (6.857), the problem set number and question, and the date. We have provided templates for L<sup>A</sup>T<sub>E</sub>X and Microsoft Word on the course website (see the *Resources* page).

**Grading and Late Policy:** Each problem is worth 10 points. Late homework will not be accepted without prior approval.

With the authors' permission, we will distribute our favorite solution to each problem as the "official" solution—this is your chance to become famous! If you do not wish for your homework to be used as an official solution, or if you wish that it only be used anonymously, please note this on your homework.

### Problem 1-1. Security Policy.

Recently, many physician practices across the country have adopted electronic practice-management systems. These computer applications allow doctors and practice employees to electronically manage many of the tasks they previously accomplished through paper documents. Doctors can document patient encounters and other medical information online. Practice billing staff can electronically submit insurance claims on behalf of patients through an online network connection with the insurance company. Patient lab appointments can be scheduled through the online system, and their lab results can be sent back to the practice via the network. Prescription requests can be electronically sent to pharmacies. Patient records can be reviewed online by the patients themselves, and can be electronically exchanged with other doctors the patient visits. However, these applications raise various privacy and security concerns. Federal legislators are concerned that these practice-management systems may cause patient health records to be revealed to others without the patient's permission. In addition, regulators are concerned that online prescription functionality may cause prescriptions to be sent to a pharmacy without actual permission from a doctor. In order to alleviate some of these concerns, federal regulators plan to set up a commission to certify practice-management applications. In order to be certified, a software vendor must provide an acceptable security policy for their product, and demonstrate that their product implements that security policy correctly.

**Your Task** You are to help the certification commission, by writing a short security policy as an example of what a practice-management software vendor should provide. Specifically, write a security policy for a practice-management system that allows medical records to be recorded online, and shared with the patient, as well as other doctors. This software is an online application that is used by doctors, patients, and other practice staff, such as receptionists, medical assistants, and billing staff. In addition, this application can be used to electronically submit prescription requests to patients' pharmacies. Your security policy should take into account the concerns of lawmakers and regulators (i.e., patient medical information being revealed to unauthorized parties, and prescriptions being sent without a doctor's permission), but should not be too restrictive.

For help on writing a security policy go to the *Resources* page on the course web site and click on *Sample Solutions from PS1 2003*. See question 1-4, which asked students to develop a security policy for either the MIT Card or Apple's iPod. Sample solutions for both, as well as a short discussion from the TAs regarding common omissions, are included. These should help guide you in terms of content, format, and length.

**Problem 1-2. Two-time Pad**

Professor Rivest told the two TA's for 6.857 before class to read up on one of his favorite papers, Diffie and Hellman's *New Directions in Cryptography*. Afterwards, each TA picked two random non-consecutive sentence from the paper, and put one after the other. Then, Professor Rivest told each TA to encrypt their two sentences with a one-time pad. Yuran encrypted his sentences, the string  $S_1$ , by taking the xor of  $S_1$ , with his pad,  $P_1$ , and ended up with the a string  $C_1$ . Jason encrypted his sentences, the string  $S_2$ , by xor-ing it with his pad,  $P_2$ , and ended up with  $C_2$ .

However, Professor Rivest later found out that Yuran and Jason were lazy, and thus copied each other's one-time pads. Thus,  $P_1 = P_2$ . Because they used the same one-time pad twice, you should be able to decrypt the original strings that Yuran and Jason chose,  $S_1$  and  $S_2$ .  $C_1$  and  $C_2$  can be found in the files enc1.bin and enc2.bin, linked from the Resources section of the course website, where you can also find a copy of *New Directions in Cryptography*.  $S_1$  and  $S_2$  are each composed of two complete sentences from that paper. Find  $S_1$  and  $S_2$ . A good solution does not need to use the length of sentences in the paper.

**Problem 1-3. Hashing**

Let  $b$  denote a given "message block size" (e.g.  $b = 512$  bits).

For this problem, assume all messages are exactly  $k$  blocks long, for some moderate  $k$  (e.g.  $k = 1000$ ). Each message has length  $bk$  bits.

Let  $n$  denote a given desired hash output size, in bits (e.g.  $n = 160$ ).

Let  $Maps(t, u)$  denote the set of all possible functions with domain  $\{0, 1\}^t$  and range  $\{0, 1\}^u$ . A randomly chosen function from  $Maps(t, u)$  may be viewed as a "random oracle" (from  $t$ -bit strings to  $u$ -bit strings).

Ideally, a hash function should be indistinguishable from a random oracle with the same domain and range. However, in practice this may not be the case, due to the manner in which the hash function is constructed.

- (a) [Birthday Paradox review] Suppose  $f$  is a random oracle drawn from  $Maps(bk, n)$ . Suppose you draw values  $x_1, x_2, \dots$  uniformly at random from  $\{0, 1\}^{bk}$ , and for each such  $x_i$  you compute  $f(x_i)$ . How many such  $x$ 's do you expect to have to try before you find a "collision" (a pair of distinct  $x_i, x_j$  values such that  $f(x_i) = f(x_j)$ )? (No need for proof here. Also, your answer does not need to be exact, just a reasonable approximation.)
- (b) Now suppose that hash function  $h$  mapping  $\{0, 1\}^{bk}$  to  $\{0, 1\}^n$  is constructed in a serial fashion from a random oracle  $g$  drawn from  $Maps(b + n, n)$ , as follows. To compute  $h(M)$  where  $M = m_1 m_2 \dots m_k$  (and each  $m_i$  is  $b$ -bits long):
- Let  $v_0 = 0^n$ .
  - Let  $v_i = g(v_{i-1} m_i)$  for  $i = 1, 2, \dots, k$ .
  - Let  $h(M) = v_k$ .

Argue that one can distinguish such a hash function  $h$  from the random oracle of part (a) having the same domain and range, by looking for collisions in a certain way.

Explain. Be quantitative.

(Hint: what happens if you just vary  $m_1$  in your tests?)