Massachusetts Institute of Technology
6.857: Network and Computer Security
Professor Ronald L. Rivest

Handout 5
September 11, 2003

Problem Set 1

This problem set is due *on paper,* in room 6-120 on *Thursday, September 18* at 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, seek partners by emailing `6.857-students-public@mit.edu`.

Homework must be typed! Each problem answer must appear on separate sheets of paper. Mark the top of each sheet with your name(s), the course number (6.857), the problem set number and question, and the date. **Homework must be typed and clear.** We have provided templates for LaTeX and Microsoft Word on the course website.

Grading and Late Policy: Each problem is worth 10 points. Late homework will not be accepted without prior approval. Homework should not be submitted by email except with prior approval. (*Somebody* from your group should be in class on the day that the homework is due.)

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. PGP

This problem has both individual and group elements to it. Your group should turn in one write-up answering each of the parts labelled [**Group**], but all key pairs, emails, etc. should be created and sent individually.

Read Alma Whitten's paper, "Why Johnny Can't Encrypt." (There is a link to it on the "Lectures and Handouts" page of the course website.)

Locate and install a fresh version of PGP or GPG. There are versions for Unix flavors, Windows, and the Macintosh. See `http://web.mit.edu/network/pgp.html` for downloads. If for some reason you are not able to download from this site, `http://www.pgpi.org/` may be of use.

Find the PGP public keys for as many of the 6.857 staff as you can. Part of your assignment is figuring out how to locate PGP keys. Searching the Internet for PGP key servers may be of help. But beware; there may be fake keys out there...

(a) [**Group**] **Reflections on Trust.** PGP's "web of trust" model allows users to "sign" each others' public keys. Suppose Alice signs Bob's key; what, in effect, is Alice declaring when she does this? Why is it useful for people to sign each other's keys? What precautions should one take before signing someone else's key, and why are these measures appropriate?

(b) [**Individual**] **Getting started.** Create a new public/private key pair for yourself (you may use an existing key pair if you already have one). Sign each of your group members' public keys, and have them sign yours.

When all of your group members have signed your public key, email it to `6.857-tas@mit.edu` in ASCII-armored format, with the subject `My public key`.

(c) [**Individual**] **Encrypting email.** Send an encrypted, signed email to `6.857-tas@mit.edu` with the subject `PGP is fun`. Do *not* send the mail to the TAs individually. In the body of the message,

- Tell us what operating system and version of PGP you are using.
- Show us the public keys you found for the 6.857 staff; PGP fingerprints are sufficient.
- In a few sentences, explain why you do or do not believe that these keys do indeed belong to the 6.857 staff. If you do not trust a public key, explain what would convince you otherwise.

Your mail should be protected with PGP such that the 6.857 TAs, and *only* the 6.857 TAs, can obtain the plaintext contents. You must also sign the mail with your private key. We will only accept your first message, so make sure to get it right the first time. Are you able to finish the assignment in fewer than 90 minutes as in Whitten's experiment? Remember to cite all your sources (books, manuals, friends, etc.) according to the guidelines in Handout 1.

(d) [**Group**] **Acting Presidential.** Find a PGP key for `president@whitehouse.gov` on a PGP key server. Based on your findings, explain one useful feature and one drawback of PGP key servers. Limit your answer to two paragraphs. Remember to cite all your sources.

Problem 1-2. Password Sniffing. [**Group**] Discuss the opportunities for password "sniffing" (eavesdropping) in *one* of the following scenarios. Consider software, hardware, network-based, and electronic means (among any others you may think up). Find at least *five* vulnerabilities.

- Alice visits her friend Bob's office. While she is there, she sits down at Bob's computer and uses it to access the MIT WebMail server. To use WebMail, Alice needs to enter her username and password, which are sent over an SSL-protected link to the web server.
- Louis Reasoner takes his laptop to the local Starbuck's to enjoy an espresso and free Wi-Fi wireless net access. He uses `telnet` to log in to his Linux machine back in his dorm room.
- Eve, sitting at home, wants to make a PPP connection over an analog telephone line to her ISP. Her ISP uses PAP authentication (see RFC 1334 for a discussion of PPP authentication protocols). The ISP's modem bank communicates over Ethernet to an authentication server that stores usernames and passwords. Eve types her username and password into a Windows dialog box, her modem calls a local phone number, and the ISP authenticates her.

Problem 1-3. Exercises in Hashing [Group]

(a) Ideally, a hash function will have the properties of both collision-resistance and pseudorandomness. Is one property stronger than the other? In other words: if a hash function is collision-resistant, must is also be pseudorandom? And vice versa? For each implication, either argue that it is true, or give a counterexample.

(b) Bitdiddle Security, Inc. proposes the following authentication scheme using a hash function h: the client and server systems both maintain a secret string s, which is initialized to some 16-word random value. Whenever the client wants to authenticate itself, the server chooses a fresh, random "challenge" string r (of length 16 words) and sends it to the client. The client sends $h(s \circ r)$ to the server. If the server receives the expected value, it accepts the client, and both the client and the server update s by appending r to it (i.e., $s \leftarrow s \circ r$). Otherwise, the server rejects the client and leaves s unchanged.

After a cursory look over pages 218-9 of her 6.857 textbook, Alyssa P. Hacker was overheard to say, "That's all well and good if h is a random oracle, but they'd better not use anything from the MD4 family..."

Explain what Alyssa meant by her comment.

(c) Bob has two hash functions, f and g. He knows that one of them is collision-resistant (and the other isn't), but he's not sure which is which. He wants to create a new hash function h which is definitely collision-resistant. Evaluate each of the following proposals, and either argue that it is definitely collision-resistant, or describe a counterexample (as usual, the $\circ$ symbol denotes concatenation):

1. $h(x) = f(x) \circ g(x)$
2. $h(x) = f(g(x))$
3. $h(x) = f(g(x)) \circ g(f(x))$

Problem 1-4. Security Policies. [Group] Write a short-but-sweet security policy for *one* of the following two scenarios. Limit your policy to one page. For example, a policy for a Project Athena workstation might specify that students should have access to the contents of their AFS lockers but should not be able to modify the system software of the workstation, while IS staff can modify the master copy of the operating system, which is then copied to the workstation by an automated process.

NOTE: It is not necessary for your security policy to be the actual security policy used by either of these organizations; we are more concerned with your ability to address all of the various aspects that need coverage, rather than with your accuracy on the specific systems in these examples.

- **The MIT Card.** As an MIT student, you are given a student ID card that contains your photograph and a magnetic strip. This card is used to open doors, check out library books, and buy food on campus. Write a security policy that governs the lifecycle of an MIT card, including the card's creation, it's use, and it's end-of-life. What are the objectives of the policy? How are different entities (students, the MIT card office, the registrar) involved?

  Make sure that your policy can handle common events, such as lost cards. Are students allowed to lend their cards to other students under your policy? Why or why not?

- **Apple's iPod.** The Apple iPod is a portable computer that contains a hard drive, a minimal operating system, a user interface, and a digital-to-analog converter. The iPod can be run under its own operating system or it can be plugged into a computer and appear as an external firewire-based hard drive. Each iPod has a unique serial number. The iPod can play music in either MP3 format or in Apple's proprietary music format, which is used to encode music downloaded from Apple's online iTunes Music Store. Apple tracks which iPods are owned by which registered iTunes users. Each iPod comes with a sticker on it that says "Don't steal music."

  Apple's goal in developing the iPod and iTunes system is to enable a service that allows users to download (and pay for) music without angering the record labels who own the music. Describe a security policy that would accomplish this goal.