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

Handout 3
February 18, 2009
Due: March 2, 2009

Problem Set 2

This problem set is due *via email,* to `6857-hw@mit.edu` on *Monday, March 2* by the beginning of class.

You are to work on this problem set in groups of three or four people. You should have received an email with your group assignment for this problem set. If not, please email `6.857-tas@mit.edu`. Be sure that all group members can explain the solutions. See Handout 1 (*Course Information*) for our policy on collaboration.

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 LaTeX 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 2-1. MAC Attack

Alice and Bob work at different branches of the super-secure Nottingham Stargazer's Association, and have need, from time to time, to send each other authenticated but non-confidential messages about their recent discoveries. They know, having taken 6.857 when they were both at MIT, that they need a "message authentication code," or MAC. They agree on the following procedure for computing a MAC of a message M, given a 128-bit key K. The procedure uses AES with 128-bit keys and 128-bit message blocks. Let $\text{AES}(K, B)$ denote the encryption of the 128-bit plaintext block B with the 128-bit key K.

The procedure first extends the message to a length that is a positive multiple of 128 bits by appending a 1 to the end of M, then appending just enough 0 bits to make its length a mutiple of 128. (This is called "padding".) Let $M_1, M_2, ..., M_n$ denote the resulting 128-bit blocks of $\text{PAD}(M)$.

To compute $\text{MAC}(K, M)$:

1. Let $V = K$
2. For $i = 1, 2, ..., n$:
   (a) Let $V = \text{AES}(M_i, V)$
3. Return V as the output value of $\text{MAC}(K, M)$

Here the message blocks M_i are used as keys in the AES calls.

(a) Do you believe that this MAC procedure is a good one? Explain why or why not.

Alice and Bob decide to add a checksum to their MAC procedure. After padding the message, the 128-bit checksum $C = M_1 \oplus M_2 \oplus ... \oplus M_n$ is appended to the message. The new message $M_1, M_2, ..., M_n, C$ is treated just like an $n + 1$ block message for the rest of the MAC procedure.

(b) Does the checksum improve the security of this MAC procedure? Why or why not?

Problem 2-2. Truncated-Key AES

Let $\text{AES}(K, M)$ denote the standard AES encryption algorithm with a 128-bit key K and a 128-bit message M.

Let $\text{AES}_s(K, M)$ denote a derived function $\text{AES}_s(K, M) = \text{AES}(0^{128-s}||K, M)$ where K is an s-bit key and $||$ denotes concatenation. Thus AES_s is similar to AES, except that it only has s-bit keys (since the first $128 - s$ bits are forced to be zero.)

Let $E_s(K, K', M)$ denote the "double encryption" of the 128-bit message M using two s-bit keys K and K': $E_s(K, K', M) = \text{AES}_s(K', \text{AES}_s(K, M))$.

Your group has been assigned a plaintext ciphertext pair (P_s, C_s) for $s = 1, 2, ..., 64$, i.e. $C_s = E_s(K_s, K'_s, P_s)$ for some unknown pair of s-bit keys (K_s, K'_s). You should have received the (P_s, C_s) pairs as an attachment to the email with your group assignment.

Find (K_s, K'_s) for the largest value of s that you can. Turn in these key values together with your program for finding them. Explain the time and memory resource usage of your program as a function of s. Is time or memory the most limiting resource?

(We have posted code for generating the (P_i, C_i) pairs to help you program calls to AES, etc. This code is posted on the course site under "Resources.")

Problem 2-3. Ben's Block Cipher

Ben Bitdiddle proposes the following block cipher. Ben's cipher operates on 128-bit input blocks and produces 128-bit output blocks.

Let $I_n = 0, 1, ..., n - 1$. A key K consists of three parts (p, S, q):

- A permutation p of I_{128}. (i.e., $p[i] \in I_{128}$ for all $i \in I_{128}$, and $p[i] \neq p[j]$ if $i \neq j$.)
- A invertible byte-substitution table S that maps I_{256} to itself one-to-one. That is, S maps every possible "input" byte to exactly one "output" byte. (Such a table is known as an S-box and is a favorite tool of cryptographers.)
- A second permutation q of I_{128}.

The block cipher E works as follows, on an input message bit vector $M[i]$, $i = 0, 1, ..., 127$:

1. The bits are permuted according to p: $R[i] = M[p[i]]$ for $i = 0, 1, ..., 127$.
2. The bits of R are grouped into bytes in $T_0, T_1, ..., T_{15}$.
3. Each byte T_i is replaced by $S[T[i]]$, yielding byte sequence $U_0, U_1, ..., U_{15}$
4. The bytes are interpreted as a bit sequence again; call this sequence $V[i]$ for i in I_{128}.
5. The bits are permuted according to q: $C[i] = V[q[i]]$ for i in I_{128}.
6. $C[0...127]$ is the output ciphertext.

(a) In Ben's scheme, there are many equivalent keys. These keys have different values for p, S, and q, but produce the same ciphertext for any given plaintext. Describe how to generate all keys equivalent to a given key (p, S, q).

(b) Describe a chosen-plaintext attack on Ben's cipher that recovers the unknown key (p, S, q) or an equivalent key.