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

Handout 4
September 18, 1997

Problem Set 2

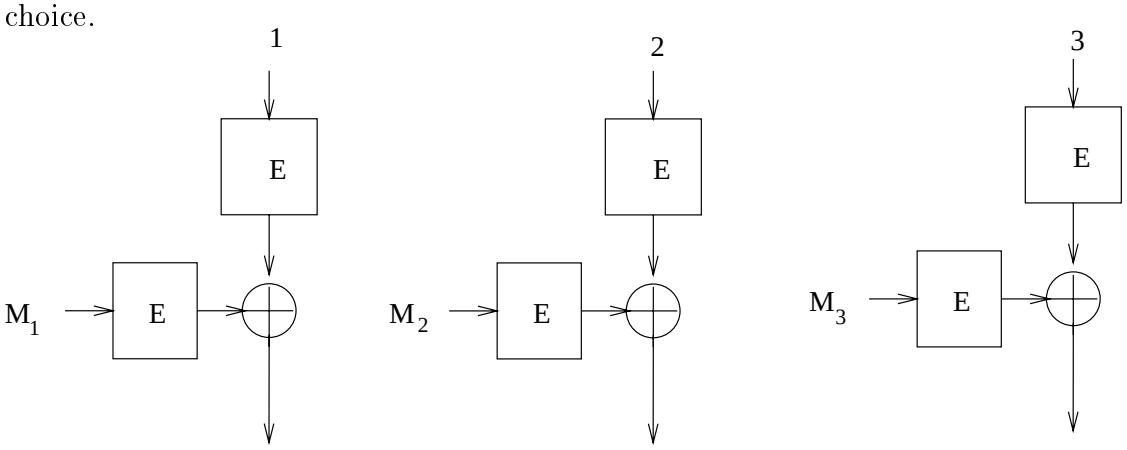
This problem set is due on *Thursday, September 25, 1997* at the end of class. Late homeworks will *not* be accepted.

Mark the top of each sheet with your name, 6.857, the problem set number, and the date. Type up your solutions, and be clear. Points may be deducted if your TA has problems understanding your solution.

If you collaborate with other students, you MUST write up solutions on your own and acknowledge the people you work with.

Problem 2-1. DES in Counter Mode for MAC

In order to analyze the security of any scheme, one first needs to define what is meant by "security." A MAC system is often defined to be "secure" if an adversary cannot create a valid message/MAC pair, after having seen the MACs of a number of other messages of his choice. In other words, a MAC scheme is insecure if, after letting our adversary request the MACs for any messages he wants, he can then construct the MAC for a new message of his choice.



Form MAC by taking low-order ten bits from each of these and concatenating them.

The following description is a MAC scheme based on the block cipher DES, which we want to analyze for security: (see figure):

- Pad the message by appending a "1" bit, and then enough zero bits to make the message's total length a multiple of 64 bits.
- Break the padded message up into 64 bit blocks.
- Encrypt each block M_x separately, yielding an intermediate ciphertext $I_x = E_k(M_x)$.

- Encrypt the number of the block and XOR it with the intermediate ciphertext to yield a final ciphertext block $C_x = I_x \oplus E_k(x)$.
- Take the low-order ten bits of each ciphertext block, and concatenate them to yield the final MAC. This makes a MAC of length approximately 1/6 of the length of the original message.

(a) Explain why the padding operation is done the way it is, rather than just padding zero bits as necessary to fill up the last block.

(b) Show that this MAC scheme is not secure, under the above definition. (Hint: truncation)

(c) Show in a different way that this MAC scheme is not secure, under the above definition. (Hint: birthday paradox)

(d) Show in a third way that this MAC scheme is not secure, under the above definition. (Hint: the adversary doesn't need to ask for any MAC's at all with this attack.)

(e) Show in a fourth way that this MAC scheme is not secure, by "cutting, rearranging, and pasting" the results from one MAC computation to yield another valid MAC. (The message for the first MAC may need to be chosen appropriately.)

(f) Which of the above four attacks still work if the XOR step is moved earlier, so that we XOR the message block with its number, encrypt the result, and take the low-order ten bits? (Explain briefly as necessary.)

(g) Which of the above four attacks still work if the XOR is moved to the key input, so that the x-th message block is encrypted with key $k \oplus x$?, and then the low-order ten bits of the result are used? (Assume that DES acts suitably "randomly".)

Problem 2-2. Message Authentication Codes versus Message Digests

Sometimes it is useful to have a short "message digest" of a long message, so that the digest can represent or "stand for" the longer message. A message digest function computes the message digest for a message; sometimes a message digest function is called a "cryptographic hash function." An essential property of a secure message digest function is that it should be computationally infeasible to come up with two messages have the same message digest.

Consider a message digest function that is essentially the DES CBC-MAC defined in class. Let K_0 denote a fixed public DES key and let IV_0 denote a fixed public initialization vector. Pad the message as is done in the previous problem. Encrypt the padded message with CBC mode using key K_0 and initialization vector IV_0. Output the final block C_n of the resulting ciphertext as the message digest of the message.

Show that it is very easy to pick two messages having the same message digest, so that the proposed use of a MAC function as a message digest function is not secure.