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

Handout 4
March 9, 2015
Due: March 23, 2015

Problem Set 3

This problem set is due on *Monday, March 23* at **11:59 PM**. Please note our late submission penalty policy in the course information handout. Please submit your problem set, in PDF format, **on Stellar**. *Each problem should be in a separate PDF.* Have **one and only one group member** submit the finished problem writeups. Please title each PDF with the Kerberos of your group members as well as the problem set number and problem number (i.e. *kerberos1_kerberos2_kerberos3_pset1_problem1.pdf*).

You are to work on this problem set with your assigned group of three or four people. Please see the course website for a listing of groups for this problem set. If you have not been assigned a group, 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: All problems are worth 10 points.

With the authors' permission, we may 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 in your profile on your homework submission.

Our department is collecting statistics on how much time students are spending on psets, etc. For each problem, please give your estimate of the number of person-hours your team spent on that problem.

Problem 3-1. Modes of operation

In class, we defined the notion of IND-CCA security. A weaker notion of security is the IND-CPA security (indistinguishability under chosen plaintext attack), in which the adversary has black-box access only to the encryption algorithm. More precisely,

Phase I:

- Adversary given black-box access to E_k (encrypt whatever it likes, without knowing the secret key k)
- Adversary outpus two messages m_0, m_1 of same length plus state information s.

Phase II:

- Examiner secretly picks $d \leftarrow \{0, 1\}$
- Examiner computes $y = E_k(m_d)$
- Adversary given y, s, access to E_k
- Adversary computes of a while, then must produce bit $\hat{d}$ as its guess for d

An encryption scheme is secure against CPA attack if $|P(\hat{d} = d) - 1/2|$ is negligible.

(a) Assume that we have a IND-CPA secure block cipher and consider the modes of operation ECB, CBC (with a random IV) and UFE for that block cipher. Which of these modes are IND-CPA secure and which are not? Argue about each case.

Modes of operation are used to encrypt various length messages. However, if the length of the message is not a multiple of the block size then in some cases we need to pad it. In the rest of the exercise, we will investigate attacks that are based on bad padding. We will assume that the adversary has additional access to a padding oracle. The adversary gives to the oracle a ciphertext and the oracle responds 1 if the corresponding plaintext is correctly padded and 0 otherwise.

(b) Assume that the padding we use is 10*, namely a 1 bit followed by as many 0's as necessary in order for the padded message to have length multiple of the block size. Is CBC mode secure against this new kind of adversary? Prove your claim.

(c) Assume that the message and the block size are counted in bytes. If the padding that we have used was 1 byte (0x80) followed by as many 0 bytes (0x00) as necessary, show how, given a ciphertext c and the size of the message in bytes n, we can retrieve the plaintext using polynomial (in the size of the message) number of accesses to the padding oracle.

(d) Can this attack be used in the case of part (b)?

Problem 3-2. Stream Ciphers

Let n be a fixed positive integer (e.g. $n = 128$) and let $X = \{0,1\}^n$.

Let f be a mapping from X to X. Here it is hoped that f has good cryptographic properties, but it may not.

In any case, we wish to define a sequence using f, for possible use in a stream cipher. Let the sequence be denoted x_0, x_1, x_2, ...

Here are four possible ways to use f to do so. (Here "i" is interpreted as an n-bit quantity when appropriate.)

(A) $x_i = f(i)$ [This is standard "counter mode"]

(B) $x_i = f(x_{i-1}) \oplus i$

(C) $x_i = f(x_{i-1} \oplus i)$

(D) $x_i = f(x_{i-1} \oplus i) \oplus i$

Define the "quality" of a sequence as the minimum, over all choices of f and all starting values x_0, of the number of distinct values appearing in the sequence x_0, x_1, ... Denote this as $Q(f)$, or $Q(f, A)$, $Q(f, D)$ for a construction using f of one of the four types above.

Show that three of the constructions above has constant $\Theta(1)$ quality, and that the other one has quality $\Theta(\sqrt{2^n})$.

Problem 3-3. AES Distinguisher

This problem relates to testing the randomness of reduced-round versions of AES. Let $AES_r(K, M)$ denote the result of encrypting 128-bit message M with 128-bit key K and r rounds of AES. (The last round uses the "final round" routine as usual for AES.) The inputs and outputs of AES are 16-byte values; let these values be subscripted 0, 1, ..., 15 (across rows, and then down columns). This problem assumes you have access to (or can create) an implementation of AES_r.

(a) Let X and Y be discrete random variables. Choose and describe briefly an effective test for telling whether X and Y have the same distribution. You may draw a sample $\{x_1, x_2, \ldots, x_n\}$ of n values according to distribution X, and similarly you may draw a sample $\{y_1, y_2, \ldots, y_n\}$ of n values according to distribution Y. You may, if desired, assume that ranges of X and Y have only a modest number of distinct possible values (e.g. 256).

(b) Let $F(r, p, q)$ denote the result (a random variable) of the following experiment; this result will be an integer in the range 1 to 256. First, pick a value M from $\{0,1\}^{128}$ uniformly at random. Let S denote the set of 256 possible messages obtained by varying byte p of M in all possible ways. Draw a 128-bit K uniformly at random. Let T denote the set of 256 ciphertexts obtained by encrypting the 256 values of S using r-round AES with key K. Let $F(r, p, q)$ denote the number of distinct values appearing in byte position q of the values in T.

See if you can distinguish r-round AES from 10-round AES by running the following experiments, for $r = 1, 2, \ldots$. For the highest r that you can distinguish, argue the correctness of your distinguisher.

- Define random variable X as $F(r, 3, 10)$. (Here we have picked $p = 3$ and $q = 10$ arbitrarily.)

- Define random variable Y as $F(10, 3, 10)$.
- Use the procedure you gave in your answer to part (a) and a fair amount of computer time to see if you can distinguish AES_r from AES_{10}. How many rounds are needed for AES_r to start looking random from the point of view of your test? Describe your results carefully.

An example AES implementation can be found at
`http://brandon.sternefamily.net/wp-content/uploads/2007/06/pyAES.txt`