

## Problem Set 5

**Due by *start* of class, Monday, April 28, 2008.**  
**(E-mail submissions O.K. if necessary)**

**Problem 1.** (Courtesy of Leo Reyzin) Read the description of CBC MAC in section 7.7 of the BR notes.

a) [10 points] Show that CBC MAC is insecure for variable-length messages. In other words, show a concrete attack on CBC MAC that asks a few queries and then outputs a forgery with high probability of success. Note that lengths of queries and forgery don't have to match in your attack, but they all have to be multiples of the block size  $n$ .

b) [10 points] Consider the following attempt at fixing this problem: prior to computing the MAC on a message  $m$ , always append its length  $\ell = |m|$  as the last  $n$ -bit block to get new message  $m' = m\|\ell$ , and then CBC MAC  $m'$ . Show that this doesn't work, either. (Hint: let  $a, b, c$  be three  $n$ -bit blocks, and let  $d$  be the  $n$ -bit block representing the integer  $n$ . Query  $a, b$  and  $a\|d\|c$ . Now figure out a new message whose tag you already know).

*Note:* Prepending (instead of appending) the length works. So does doing nothing with the length of the message, but simply passing the final encrypted block through a PRF with an independent key  $K'$  (thus, the MAC key becomes  $(K, K')$ ).

**Problem 2.** (Stallings) [10 points] Describe how using the ciphertext feedback mode of operation to generate a CFB MAC scheme similar to CBC MAC. Argue/prove that it is a secure MAC.

**Problem 3.** (Stallings) [10 points] Consider the following MAC scheme based on fixed, unkeyed hash function  $H$ . The MAC key generation algorithm selects a random key  $K$  and for message  $m$ ,  $tag \leftarrow H(K\|m)$ . To verify, the receiver recomputes the MAC and checks that the result  $tag'$  matches  $tag$ . The author claims this is a secure MAC. Why is the proposed scheme secure and which of the three hash function properties described in the BR notes: collision-resistance, one-wayness/target-collision resistance, universality must  $H$  possess for this to be the case? Explain your reasoning.

**Problem 4.** (Stallings) In addition to security properties, hash functions  $H$  have the following functional requirements:

1.  $H$  can be applied to messages of (practically) any length
2.  $H$  produces a fixed-length output
3.  $H(m)$  is relatively easy to compute for any  $m$ , so that both hardware and software implementations are practical.

a) [10 points] Consider the following hash function. Messages are written in the form of a sequence of decimal numbers,  $m = (a_1, a_2, \dots, a_t)$ . The hash function  $H_1$  is defined as

$$H_1 = \left( \sum_{i=1}^t a_i \right) \bmod n$$

for a predefined parameter  $n$ . Does this function satisfy any of the functional or security requirements for hash functions? Explain your answer.

b) [10 points] Same question as the previous part for the function  $H_2$  defined as

$$H_2 = \left( \sum_{i=1}^t (a_i)^2 \right) \bmod n.$$

c) [10 points] Calculate  $H_1(m)$  and  $H_2(m)$  for message  $m = (189, 632, 900, 722, 349)$  and  $n = 989$ .