

CAS CS 538. Problem Set 9

**Due by 5pm on Thursday, December 11, 2008, by email to
reyzin@bu.edu or under the door of MCS 287.**

In the two problems below, suppose $\{F_k\}$ is a pseudorandom function family from n -bit inputs to n -bit outputs, with a n -bit key k . Also, let \circ denote concatenation, \bar{x} denote the bit-by-bit negation of x , and 0^n denote the string of n zeroes.

Problem 1. (40 points, 10 for each part) We are trying to construct a function family with $2n$ -bit outputs out of $\{F_k\}$. Which of the following are pseudorandom function families? Prove your answers.

(a) $F_k^1(x) = F_k(0^n) \circ F_k(x)$.

(b) $F_k^2(x) = F_k(x) \circ F_k(\bar{x})$.

(c) $F_k^3(x) = F_{0^n}(x) \circ F_k(x)$.

(d) $F_k^4(x) = G(F_k(x))$, where G is a length-doubling PRG.

Problem 2. (20 points) Is the following a secure MAC? Prove your answer.

Key generation just picks k of length n . The tagging algorithm on message $m = m_1 \dots m_l$, where m_i is a n -bit block, computes $t = F_k(m_1) \oplus \dots \oplus F_k(m_l)$. The verification algorithm recomputes the tag and checks if it's correct.

Problem 3. (20 points) In class we showed that two-round Luby-Rackoff is not a pseudorandom permutation by exhibiting a two-query distinguisher. Show that three-round Luby-Rackoff, as described in the lecture notes, is not super pseudorandom. Namely, exhibit a three-query distinguisher (two forward queries followed by one reverse query), and explain why it distinguishes with high probability. (Hint: recall that our in-class distinguisher found a way to XOR a predictable value into S by varying L . You can similarly XOR a predictable value into S with a reverse query (think about how). Using this, you can get S in a reverse query to be equal to S in a different forward query. This will lead to non-random-looking relationship among the inputs and outputs.)

Problem 4. (20 points) Recall that in class we showed that CBC MAC is insecure for variable-length messages, but that *prepending* the message length works. Show that *appending* the message length does not work. (Hint: let a, b, c be three n -bit blocks, and let d be the n -bit block representing the integer 1. Query a, b and $a \circ d \circ c$, where \circ denotes concatenation. Now figure out a new message whose tag you already know).