COMP 4109: Applied Cryptography
Assignment 1

January 10, 2005

The answers to these questions are due January 19th in class. Turn in paper copies either in class or at HP 5137. In addition, please email program code and outputs to Mohammad Mannan (mannan@ccsl.carleton.ca).

1. A company claims that their new cipher is provably secure. You find out from other sources that they use a fixed 10,480-bit key. Do you believe their claim? Why?

2. What is the minimum linear complexity of a fixed 1000-bit binary sequence?

3. Please answer the following questions regarding the number π.

   (a) What is the linear complexity of a binary sequence representing the fractional part of the number π represented in binary?

   (b) Write a program that performs the Poker test (Section 5.4.4) on a binary number. Then, run the Poker test using $m = 4$ on S, where S is the first 10,000 bits of the fractional part of π (represented in binary). (You may use more bits if you wish.)

   If S was a random sequence, then it should approximate the χ^2 distribution with 15 degrees of freedom. From Table 5.2, this means that $P(\text{Poker}(S, m = 4) > 24.9958) = 0.05$. Based on this relationship and the discussion in Section 5.4.2, does S pass the length 4 Poker Test for randomness with 95% confidence? Explain.

   (c) Would π be a good choice for a keystream for a Vernam cipher? Why or why not?

4. RC4 is a simple, fast, but surprisingly secure stream cipher. Using the description of RC4 at the end of this assignment, answer the following questions:

   (a) What is the maximum key size for RC4 in bits?

   (b) Implement a modified version of RC4 which halves the maximum key size. Show a test encryption and decryption using a five byte key and a 100 byte plaintext

   (c) Will your implementation interoperate with the standard RC4? Why or why not? Be specific.

   (d) Would you be willing to recommend that a commerical bank use your modified version of RC4? Note that they require an algorithm with at least a 128-bit key. Explain your reasoning.

Description of RC4

To initialize RC4, first fill a 256-element byte array S linearly as follows: $S_0 = 0, S_1 = 1, \ldots, S_{255} = 255$. Fill a second 256-byte array K with the key, repeating the key as necessary to fill the entire array: $K_0, K_1, \ldots, K_{255}$. Set the index j to zero. Then:

for i = 0 to 255:
  $j = (j + S_i + K_i) \bmod 256$
  swap S_i and S_j

After initialization, set two counters i and j to zero. To generate a byte of the keystream, do the following:

$i = (i + 1) \bmod 256$
$j = (j + S_i) \bmod 256$
swap S_i and S_j
$t = (S_i + S_j) \bmod 256$
$k = S_t$

To encrypt a byte of plaintext, XOR it with k to produce a byte of ciphertext. To decrypt, XOR k with the corresponding byte of ciphertext.