2ND MIDTERM, MATH 587/CSCE 557 - APRIL 3, 2007

Nigel Boston

Answer all three questions below. Show your working. Full credit will not be given for just the answer without any justification. Make sure you answer each part of each question. Do not write essays! Concise, "to-the-point" answers are favored.

1. (a) The following is a Vigenere ciphertext coming from encrypting an English plaintext: ZIHLL WYGSM FAVRE PFBWP ZIHVX FGBVJ JZBRW ZCAFT UMNRO QCRGP TNBVP BFVDP UBNXX FGBVJ JMJLL UGNOP TIHVW

Explain why the Kasiski test works and use it to guess the keyword length.

Answer: If a long string of letters is repeated in two places, then they're likely to come from the same portion of plaintext matching the same portion of keyword, in which case the distance between the repetitions will be divisible by the length of the keyword. In the above ciphertext, XFGBVJJ appears twice at 40 apart, ZIH appears twice at 20 apart, and IHV appears twice at 60 apart, so the length of the keyword likely divides 20. (Could be $20, 10, 5, ...$).

(b) Define the index of coincidence of a message. The index of coincidence of the above ciphertext is about 0.033. Why does this tell us that the cipher is not monoalphabetic (i.e. that the keyword has length > 1)?

Answer: The index of coincidence is the proportion of pairs of letters that match. For a monoalphabetic cipher, this approximately equals the index of coincidence of English, which is about 0.066, so our cipher can't be monoalphabetic.

(c) If the keyword length is m and we take the subset of the message consisting of the 1st, $(m+1)$th, $(2m+1)$th, ... letters, then what approximately is the index of coincidence of this subset?

Answer: This is the index of coincidence of English, which is about 0.066.

(d) Explain how part (c) can be used to guess the keyword length. How would you then use this subset to find the 1st letter of the keyword?

Answer: Do this computation for several m - the correct m (and its multiples) will give you the largest answer. For each k in $0, 1, ..., 25$ compute $S_k = p_0 q_k + p_1 q_{k+1} + ... + p_{25} q_{k-1}$ where p_i and q_i are the frequencies of the ith English letter in typical English and in the subset respectively. The k for which S_k is largest gives you the right shift so the 1st letter of the keyword.

2. (a) Define entropy.

Answer: If a random variable takes n values with probabilities $p_1, ..., p_n$, then its entropy is $-p_1 \log_2(p_1) - ... - p_n \log_2(p_n)$.

(b) Suppose an alphabet has just 3 symbols A, B, and C, which arise with respective frequencies $p_A = 1/2,\ p_B = 1/4,\ p_C = 1/4$. Compute its entropy.

Answer: $-(1/2)\log_2(1/2) - (1/4)\log_2(1/4) - (1/4)\log_2(1/4) = 1/2 + 1/2 + 1/2 = 3/2 = 1.5$

(c) In a message of length 100 symbols using this alphabet, how many A's, B's, and C's should there respectively be? Suppose A, B, and C are encoded by $0, 10$, and 11 respectively. Explain why this is uniquely decodable. [Write down any string of zeros and ones and see how you can turn it back to A's, B's, and C's.] How many bits will the encoded message be? How is this related to your answer in 2(b)?

Answer: There should be about 50 A's, 25 B's, and 25 C's. If the first bit is 0, the first symbol is A - if the first bit is 1, then if the second bit is 0 (respectively 1) then the first symbol is B (respectively C). Then repeat with what's left of the message. The 50 A's encode to 50 bits, whilst the 25 B's and 25 C's encode to 50 bits each, so the encoded message is about 150 bits long. $150/100 = 1.5$ bits per symbol, and the answer to 2(b) says that this is the most the message can be compressed by.

3. (a) Explain why Linear Feedback Shift Register cryptosystems are just certain Vigenere cryptosystems in binary. What is the keyword length for an LFSR cryptosystem usually called?

Answer: The systems work the same way, adding in a repeated string of symbols one symbol at a time to the plaintext to get the ciphertext. For an LFSR the "keyword" is one period of its keystream, so keyword length is period.

(b) Consider the LFSR with the rule $x_{i+4} = x_i + x_{i+2} \pmod 2$ and starting $x_0 = x_1 = x_2 = x_3 = 1$. Give its first 10 bits. What is its period?

1111001111, period 6.

(c) If ciphertext 1001010110 is output using the keystream produced by the LFSR of 3(b), what is the plaintext?

Subtracting 1111001111 from 1001010110 one bit at a time gives plaintext 0110011001.