MATH 587/CSCE 557 - SUMMARY OF CLASS, 2/20/07

ΦI began by going over the solutions to the 1st midterm, which are posted online. I gave some hints on good practice in using frequency analysis to decipher a ciphertext created by a general substitution cipher.

We finished up the topic of chosen and known plaintexts. In the case of Viginere ciphers, for a chosen plaintext attack, input AAAA... and then read off the key as the output. In particular, if the key word has length m, then an input of m letters is enough to recover the key. For a known plaintext attack, the key is just the output minus the input, and so once again the first m letters (of plaintext and ciphertext) are enough to recover the key (by subtraction mod 26).

For Hill ciphers, look at the case of block length 2. We want a key matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ Suppose input x_1, x_2, x_3, x_4 leads to output y_1, y_2, y_3, y_4. Then $y_1 = ax_1 + bx_2, y_2 = cx_1 + dx_2, y_3 = ax_3 + bx_4, y_4 = cx_3 + dx_4$. We can write this as:

$$\begin{pmatrix} y_1 \\ y_3 \\ y_2 \\ y_4 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & 0 & 0 \\ x_3 & x_4 & 0 & 0 \\ 0 & 0 & x_1 & x_2 \\ 0 & 0 & x_3 & x_4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

Calling the 4 by 4 matrix M, we see that we can solve for a, b, c, d if M is invertible (mod 26), which is iff $x_1x_4 - x_2x_3$ is in $\mathbf{Z}_{26}^*$. This happens approximately with probability 6/13 since $\mathbf{Z}_{26}^*$ has 12 elements. Thus with probability about 6/13, 4 letters suffice for a known plaintext attack on a blocklength 2 Hill cipher. As for a chosen plaintext attack, just pick x_1, x_2, x_3, x_4 so that $x_1x_4 - x_2x_3$ is in $\mathbf{Z}_{26}^*$, so 4 letters certainly suffice here.

Then we began a systematic study of frequency analysis. Suppose that $P = (p_0, p_1, ..., p_{25})$ gives the expected frequencies of A,B,...,Z in an English passage. Let $Q = (q_0, q_1, ..., q_{25})$ be the observed frequencies of A,B,...,Z in a passage to be decrypted. For each possible key k, consider

$$S_k = (p_0 - q_{e_k(0)})^2 + (p_1 - q_{e_k(1)})^2 + ... + (p_{25} - q_{e_k(25)})^2$$

If k is the correct key, then p_0 will approximate $q_{e_k(0)}$, p_1 will approximate $q_{e_k(1)}$, and so on, and so S_k will be small. Expanding out S_k, we get:

$$p_0^2 + p_1^2 + ... + p_{25}^2 + q_{e_k(0)}^2 + q_{e_k(1)}^2 + q_{e_k(25)}^2 - 2p_0q_{e_k(0)} - ... - 2p_{25}q_{e_k(25)}$$

In particular, S_k is smallest when the correlation, $C_k = p_0q_{e_k(0)} + ... + p_{25}q_{e_k(25)}$, is largest (the sum of the squares of the p's and sum of the squares of the q's do not depend on k). Thus, one approach, if there are relatively few possible keys, is to compute C_k for each key k, find the keys for which C_k is largest and try decrypting with those. Often, the correct key is arg max C_k.