MATH 587/CSCE 557 - SUMMARY OF CLASS, 2/27/07

I recapped the theory of Kasiski's and Friedman's attacks on the Vigenere cryptosystem. I pointed out some drawbacks, such as that the Kasiski attack can ascribe importance to repetitions that occur at random. Also, the Friedman attack as given is not very sensitive. We considered refinements.

If y and z are strings of letters of equal length, say $y = y_1y_2...y_N$ and $z = z_1z_2...z_N$, define their incidence of coincidence to be $I(y,z) = (1/N)\#\{i : y_i = z_i\}$. In particular, for each positive integer m, set $y^{(+m)}$ to be the shifted sequence $y_{m+1}y_{m+2},,,$ and consider $I(y, y^{(+m)})$ (to take care of the difference in lengths, wrap around back to y_m).

The point now is that if y is a Vigenere ciphertext and m is a multiple of the keyword length, then since y_1 and y_{m+1} are enciphered by the same shift (the 1st letter of the keyword), since y_2 and y_{m+2} are enciphered by the same shift (the 2nd letter of the keyword), and so on, $I(y, y^{(+m)})$ will equal approximately 0.066 (the probability that two letters picked at random from an English text are the same).

So if we take the Vigenere ciphertext y and compute $I(y, y^{(+m)})$ for each m, usually this will be approximately $1/26 = 0.0385$, except for certain special m, which should be precisely the multiples of the keyword length. An example was given where the keyword length is clearly identified this way.

Suppose we have the keyword length - how to find the keyword?

Suppose the Vigenere ciphertext is y and the keyword length is known to be m. As noted above $y_1, y_{m+1}, y_{2m+1}, ...$ arise by applying the same shift (the 1st letter of the keyword) to letters whose frequency distribution should match that of English. Let the 26-tuple $\mathbf{b} = (b_0, b_1, ..., b_{25})$ denote the typical frequencies of A,B,...,Z respectively in English. Similarly let the 26-tuple $\mathbf{a}_1 = (a_{1,0}, a_{1,1}, ..., a_{1,25})$ denote the frequencies of A,B,...,Z respectively in the set $\{y_1, y_{m+1}, y_{2m+1}, ...\}$. Then as noted when talking about fancy ways of decrypting a shift cipher, if the first letter of the keyword is k_1, then the $(i + k_1)$th entry of a_1 should approximate the ith entry of b. We therefore compute, for each k, $a_{1,k}b_0 + a_{1,k+1}b_1 + ... + a_{1,k-1}b_{25}$. The k for which this is largest is our guess of k_1.

To get the second letter of the keyword, we do the same for $\mathbf{a}_2$, which counts the frequencies of A,B,...,Z respectively in $\{y_2, y_{m+2}, y_{2m+2}, ...\}$. Checks on this process are usually whether the keyword is a legitimate English word (not required, but usually the case) and if subtracting the keyword repeated from the ciphertext yields English that makes sense.

Next time, we'll use this to solve a specific Vigenere challenge ciphertext.