Vigenere Cipher

Although the form of this cipher that is currently in most books isn't exactly what Vigenere proposed, it shows the essential idea that Vigenere had to thwart frequency analysis.

The idea is as follows:

Pick a keyword, such as "COMPUTER". Now, to encrypt a message, do the following:

```
Plaintext:  MEETMEATTHESTORE
Keyword  :  COMPUTERCOMPUTER
Ciphertext: OSQIGXEKVVQHNHVV
```

Essentially, to encrypt, you line up the plaintext with the keyword written down repeatedly (until you get to the end of the message), and then you add the numeric values of both letters (0 to 25) and take the result mod 26 to obtain the corresponding ciphertext letter. In Java code, we'd have something like this:

```
for (i=0; i<msg.length; i++)
  cipher[i] = (char)((msg[i]-'a'+key[i%key.length]-'a')%26+'a');
```

In python we might do this:

```
for i in range(len(msg)):
    numC = (ord(msg[i])+ord(key[i%len(key)])-2*ord('a'))%26
    cipher.append(chr(numC+ord('a'))
```

Decrypting is performed by simply subtracting out the values of the keyword from the ciphertext in the appropriate manner.

It is clear that this cipher disrupts frequency analysis because the same letter in the ciphertext (such as the two Vs at the end) can be obtained from two different letters in the plaintext. Similarly, two of the same plaintext letters can map to different ciphertext letters.

Of course, if one knew the key length, then one could obtain some frequency information, by picking at every kth character, where k was the key length. (In essence k groups of characters could be formed. Each of the letters within a single group would have been shifted by the same key letter, thus, frequency information of each group would be preserved in these ciphertext letter groups.