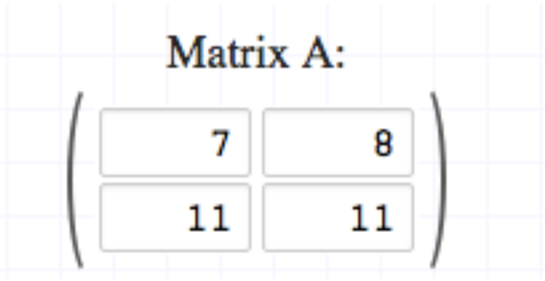
Spencer Fajardo
Math 2200
Hill Cipher

The hill cipher is an interesting cryptographic method invented by Lester S. Hill in 1929. This particular Cipher uses elements from Linear Algebra in order to scramble messages, or encrypt them. In addition, the hill cipher also makes use of modular arithmetic. We shall take a look at how matrices are used to hold information about the phrases we wish to encrypt, and how inverse matrices are used in order to decipher the scrambled message.

In order for the hill cipher to work, we need a way to numerically code the letters into a matrix. For this purpose, we set the letter a to zero, the letter b to one, and so on until z is at 25. Now, we can simply replace the letter with the corresponding value every time we need to put information into a matrix.

The hill cipher has two main components to it. The first is the key. The key in this project is a four-letter word used to encode the message. The letters of the key are placed into a 2x2 matrix, using the numerical values of the letters. For example, if our key is ‘hill’, the corresponding matrix would be;



Matrix A:

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}$$

It is important to note that the key we choose needs to have a matrix representation with a non-zero determinant for the encryption to work properly.

Now that we have our key, the other component of the hill cipher is the phrase we wish to encode. The phrase can be of any length, but the hill cipher only works on groups of two letters, thus if the phrase is odd, a junk character needs to be added to the end of the phrase. For simplicity, we will only use even-lettered phrases. The phrase is put into a series of 1x2 vectors. For example, if our phrase is ‘cats’, the vectors would be:

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix}\begin{pmatrix} 19 \\ 18 \end{pmatrix}$$

Notice how the letter c corresponds to the number 2 in the first vector and a is the second number in the first vector.

The next step is to use the key matrix and the phrase vectors to compute a new, scramble message. This is done in multiple steps. The first step is, for each vector, to create a new 2x1 vector by multiplying it against the key matrix. Using our example, the first vector multiplied against the key matrix becomes the vector:

$$\begin{pmatrix} 14 \\ 22 \end{pmatrix}$$

And the second vector becomes:

$$\begin{pmatrix} 277 \\ 407 \end{pmatrix}$$

This second vector has values beyond the range of the alphabet. Since we don't have a 277th letter or a 407th letter, we need a way to convert these values back into the range 0-25. To do this, we use modular arithmetic. Since the alphabet is 26 letters long, we divide the values by 26 until we reach the remainder, and we simply keep the remainder. So, 277 mod 26 becomes 17 and 407 mod 26 becomes 17 as well. Now, our second vector is

$$\begin{pmatrix} 17 \\ 17 \end{pmatrix}$$

Which is again within the range of the alphabet. So, using matrix multiplication, we have converted our phrase 'cats' into the phrase 'owrr', which is nicely encoded.

So far, we have put our key and phrase into a matrix and vectors, and we have computed an encoded message using matrix multiplication. Now we need to decipher our message, and to do this we need to find the inverse of our key matrix. To find the inverse, we need two things. We need the multiplicative inverse of the key-matrix's determinant mod 26, and we need the adjugate matrix of the key. Consider a matrix of the form:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Then, the determinant of the matrix is a*d – b*c, which is easy to compute. Now, to find the inverse of the matrix mod 26, we need to find a number k such that the determinant of the matrix det(A) * k = 1 mod 26. That is, the determinant of the matrix times k equals 1 when you take the remainder mod 26. To find this k, you simply loop through 0 – 25 and test which value times the determinant is equal to 1 mod 26. Now we need to find the adjugate matrix, which, if we consider the matrix above, looks like this;

$$\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

With the adjugate matrix and the multiplicative inverse k, we multiply the adjugate by the multiplicative inverse and take the result mod 26. In the case of our example we get

$$\begin{pmatrix} 11 & 18 \\ 15 & 7 \end{pmatrix} \cdot 7 = \begin{pmatrix} 25 & 22 \\ 1 & 23 \end{pmatrix} mod\, 26$$

Notice that we took the adjugate matrix mod 26 before we multiplied. To decipher our message, we take the new inverse matrix we just computed and we multiply it by the encrypted vectors we created earlier, and in doing so we get back our original message. Thus, we have used matrix representations of words and phrases to create an encrypted message, and we used the inverse matrix to decipher the message. Our message would have been nearly indecipherable if we did not have access to the original key, and that is where the power of the hill cipher lays.

In conclusion, we have shown a way that you can numerically represent words and phrases as matrices, and used matrix properties from linear algebra to compute encoded messages that are only decipherable if you have access to the key used to encode the message. Some of the positive aspects of the hill cipher is the simplicity in which messages are encoded, and how easy it is to compute the necessary components to decipher the message. One of the drawbacks of the hill cipher is that you need to find a key that has a non-zero determinant as a matrix which can take some trial and error. But, if you are comfortable with modular arithmetic and basic matrix properties, a fun cipher can be created with little effort if you have access to a computer. As part of this project, I have included java code that takes in even-lettered phrases and a key, and created encoded messages, and also deciphers messages as needed.