# Hill Cypher

The Hill cipher algorithm is a symmetric key algorithm which means that we can get the decryption key out of the encryption one with a simple transformation.

The idea behind hill cipher algorithm is actually pretty simple. It takes ***m*** successive plain pixels and substitutes them with ***m*** ciphered pixels. For example, if we have three pixels P1, P2, P3, Hill cipher algorithm will substitute them by C1, C2, C3 according to the following equations:

C1 = (K11 P1 + K12 P2 + K13 P3) mod range

C2 = (K21 P1 + K22 P2 + K23 P3) mod range

C3 = (K31 P1 + K32 P2 + K33 P3) mod range

This can be represented by matrices as following:

This can simply be written as: **C = K P**

Where **P** is the block of plain pixels that is to be substituted by **C** (block of encrypted pixels). **K** is the key matrix that will do this transformation.

To decrypt the image: **P = K-1 C**

As you have probably noticed, the key matrix must be invertible so that we can decrypt the image back.

The main concept of Hill cipher algorithm is pretty clear. The only thing that we need to work on is how to find the key matrix **K** so that we guarantee that it is invertible. We will discuss this in the next section.

## Generation of Involutory Key Matrix

In the algorithm we use, The generated key matrix for Hill cipher algorithm will be involutory, which means that the key matrix is its own inverse e.g. **K2 = I**

Let **A** = be involutory matrix

Matrix **A** will be partitioned into 4 smaller matrices each of order

**A** =

Since **A2  = I**, then we can deduce the following:

* A12 A21 = I – A112  = (I – A11) (I + A11) (Equation 1)
* A11 + A22 = 0 (Equation 2)

So, to generate an involutory matrix **A**:

* A22 can be any matrix, so we fill it randomly.
* From Equation 2: **A11 = -A22**
* Let **A12 = K(I – A11)** or **K(I + A11)** (Equation 3)

From Equation 1 and 3: **A21 =** or

### Results



Figure 2: Original Image (2)

Figure 1: Original Image (1)

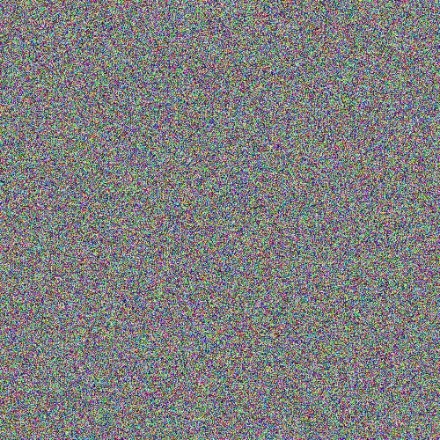


Figure 6: Decrypted Image (2)

Figure 4: Encrypted Image (2)

Figure 3: Encrypted Image (1)

Figure5: Decrypted Image (1)

We can see that the encrypted image is bigger than the original one as we change the image to have square dimensions, so we can multiply it by our Involutory Key Matrix.  
The added pixels have Zero RGB values.