WHAT DOES A GIVEN MATLAB FILE DO?

if you want to know what does a given matlab file do, type help matlab\_file\_name (without .m extension)

e.g. to know what arrange\_by\_pattern.m does,type

help arrange\_by\_pattern

STEPS TO WORK OUT THE PROJECT

1. Open read\_image.m file and change the path to your image's path.
2. image = read\_image();

this will read the image from the path you mentioned in first step.

1. original\_size = [size(image,1) size(image,2)];

original\_size is saved to get back original image at the very end of decryption process.

1. image = preprocess\_image(image);

this will preprocess the image as required for encryption.

read its details by help preprocess\_image

1. key = 'X3(B2(d3,o4,s5,c6), s3, d2, Z0(d2, o3, s4, c5))';

this is the key for encryption and decryption process.

you can design your own key which will be told to you later

1. crypted\_image = crypt\_image(image, key);

this encrypts the image.

you can view the crypted image by figure,imshow(crypted\_image);

1. decrypted\_image = decrypt\_image(crypted\_image, key);

this decrypts the image.

1. original\_decrypted\_image = get\_original\_image(decrypted\_image, original\_size);

this is the final decrypted image

view it by figure,imshow(original\_decrypted\_image);

IT WORKS, YIPEEEE!!!!

1. (optional step) if you want to know correlation between image and crypted\_image

compute\_correlation\_grayscale(image, crypted\_image)

HOW TO GENERATE THE KEY

* this concept works on grayscaled square shaped image

we add up black space to convert the given rectangular image into square image.

suppose your image was of size 460\*600

so we change it to 600\*600

* we can divide only even dimension square

i.e. 5\*5 image can't be divided

but 6\*6 can be divided

600 can be divided to 300 sized subimages

300 can be further divided to 150 sized subimages

150 can be further divided to 75 sized subimages

no more division is allowed

* to divide an image into 4 subimages,we need pattern\_key

pattern\_key can be B,Z or X

it has to be accompanied by a number from 0 to 7 telling the rotation

e.g. are B0,Z5,X7,etc

pattern\_key has to be always accompanied by four scan\_keys separated by , and enclosed in ()

scan\_key can be c,o,s,d again followed by a rotation number from 0 to 7

valid key is B6(o3,d1,c6,s2)

keys can be further made complex by nesting

nesting can be done as replacing any 1 of the scan\_key with a new pattern\_key

e.g. B6(o3,d1,c6,s2) can be nested as B6(o3,d1,Z2(c1,o2,d3,s4),s2)

nesting is only allowed till number of steps for which division is allowed

e.g. 2 times nesting is allowed in a key for 600\*600 image

whereas 6 times nesting is allowed in a key for 256\*256 image

It all depends on upto what point you avoid getting an odd number in dividing the size by 2