

```
1 clear; clc;
2
3 %Reading DATA_OUT from the my_random_numbers.m file into rand_nums array
4 fileID = fopen('my_random_numbers.m','r');
5 formatSpec = '%f';
6 sizeA = [1 inf];
7 rand_nums = fscanf(fileID,formatSpec,sizeA);
8
9 %Opening the input image and converting it to a 3D array of pixels named A
10 A = imread("my_image_2.jpg");
11 image(uint8(A));
12 pause;
13 R_matrix = A(:,:,1); G_matrix = A(:,:,2); B_matrix = A(:,:,3);
14
15 %Initializing the RAND_matrix and A_encrypted arrays
16 [rows,cols,depth] = size(A);
17 RAND_matrix = zeros(rows,cols,depth);
18 A_encrypted = zeros(rows,cols,depth);
19
20 %Encrypting the image
21 %Iterating through the RAND_matrix and storing a value of rand_nums
22 %XORing the current indexed value of RAND_matrix and A, into A_encrypted
23 c = 1;
24 for i = 1:rows
25     for j = 1:cols
26         for k = 1:depth
27             if c == (width(rand_nums))
28                 c = 1;
29             else
30                 c = c + 1;
31             end
32             RAND_matrix(i, j, k) = rand_nums(c);
33             A_encrypted(i, j, k) = uint8(bitxor(A(i,j,k),
34 RAND_matrix(i,j,k)));
35         end
36     end
37 end
38 %Displaying the encrypted image
39 image(uint8(A_encrypted));
40 pause;
41
42 %Initializing A_decrypted array
43 A_decrypted = zeros(rows,cols,depth);
44
45 %Using the same steps as to encrypt, the image is decrypted
46 c = 1;
47 for i = 1:rows
48     for j = 1:cols
49         for k = 1:depth
50             if c == (width(rand_nums))
51                 c = 1;
52             else
53                 c = c + 1;
54             end
55             RAND_matrix(i, j, k) = rand_nums(c);
56             A_decrypted(i, j, k) = uint8(bitxor(A_encrypted(i,j,k),
57 RAND_matrix(i,j,k)));
58         end
59     end
60 end
```

```
58     end
59 end
60
61 %Displaying the decrypted image
62 image(uint8(A_decrypted))
```