

Design Credit Report

Work on Coloured Images using Visual
Cryptography

Anirudh Srikanth (B20CS006) & Sumit Kumar Prajapati (B20CS74)

Under Prof. Somitra Kumar Sanadhya

1 May 2022

Background

Cryptography is the study of secure communication techniques that only allow the sender and the recipient to view its contents. The term is derived from the greek word *kryptos*, which means hidden.

Visual Cryptography is a cryptographic technique where images are distributed as shares, and the recovery of the final image can be done using the human visual system, usually by having the shares be printed on transparencies and having them stacked on top of other.

Our Work

Basic Visual Cryptography

We first implemented the Basic Visual Cryptography Scheme Made by Moni Naor and Adi Shamir based on this [paper](#).

The scheme involves a (n,k) secret sharing scheme, where the image is divided into n shares, such that any $k-1$ shares or fewer when stacked on top of each other must be unable to view the image. And any k or more images when stacked on each other should be able to view the image completely.

The scheme would be perfectly secure and very easy to implement. The method exploits the human visual system, which can detect the contrast of images, hence a grid with more black pixels would appear more black than a grid with more white (transparent) pixels.

The original method involved binary images. i.e each pixel was either on or off. In this scheme each pixel was mapped to another vector of pixels of size m which are closely stacked together. And based on whether the original

pixel was white or not, we can fill the mapped vector of pixels by exploiting the concept that if the hamming weight, $H(V) \geq d$ the Human Visual system would see it as black, and if the $H(V) < d - \alpha m$ for some fixed threshold $1 \leq d \leq m$ and relative difference $\alpha > 0$.

Now based on these constraints we can make two boolean matrices C_0 and C_1 where they are of shape $n \times m$, where each row can be distributed as a share, and the or of the rows satisfy the above requirements, C_0 for white and C_1 for black.

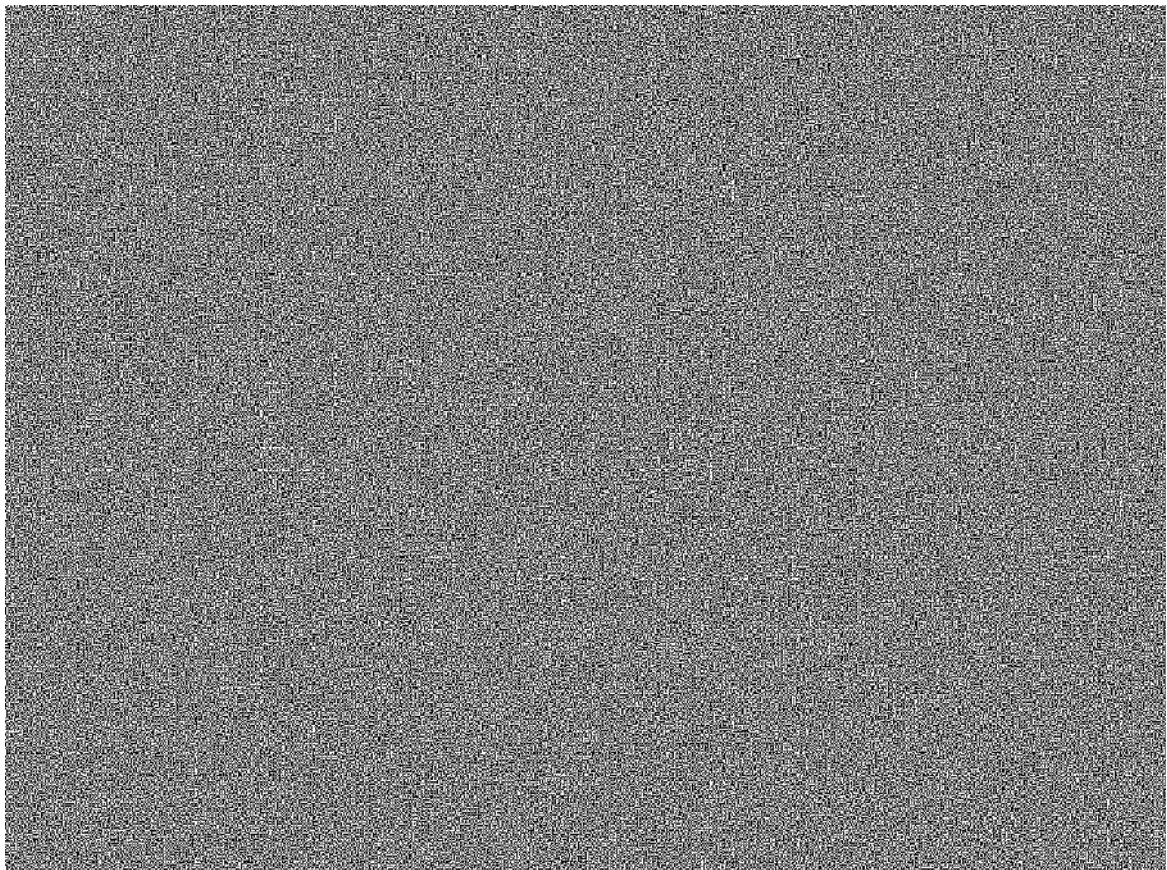
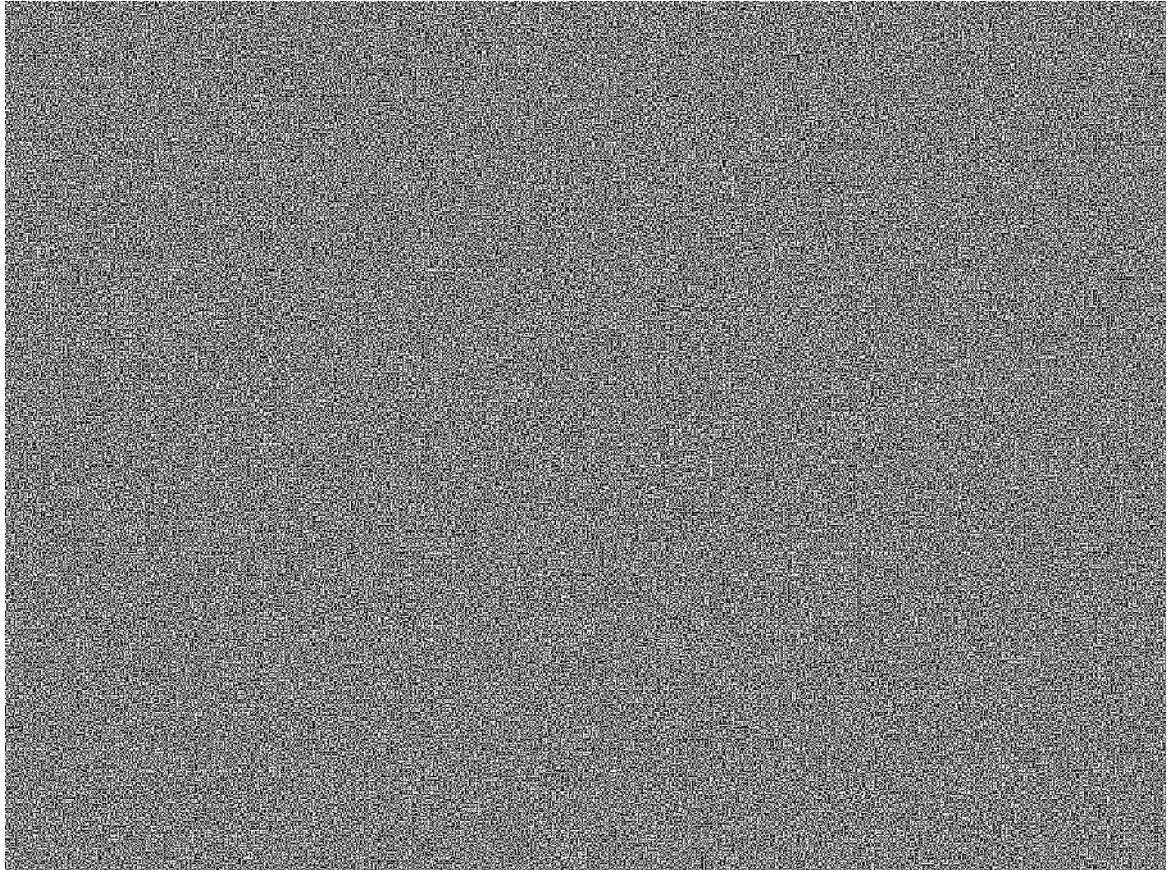
Now while we made the map for each pixel, we shuffled the columns, to increase the security of the scheme.

Here are the results of the (2,2) scheme that we implemented, i.e a scheme where an image was distributed as 2 shares, and needs both shares to see the original image.



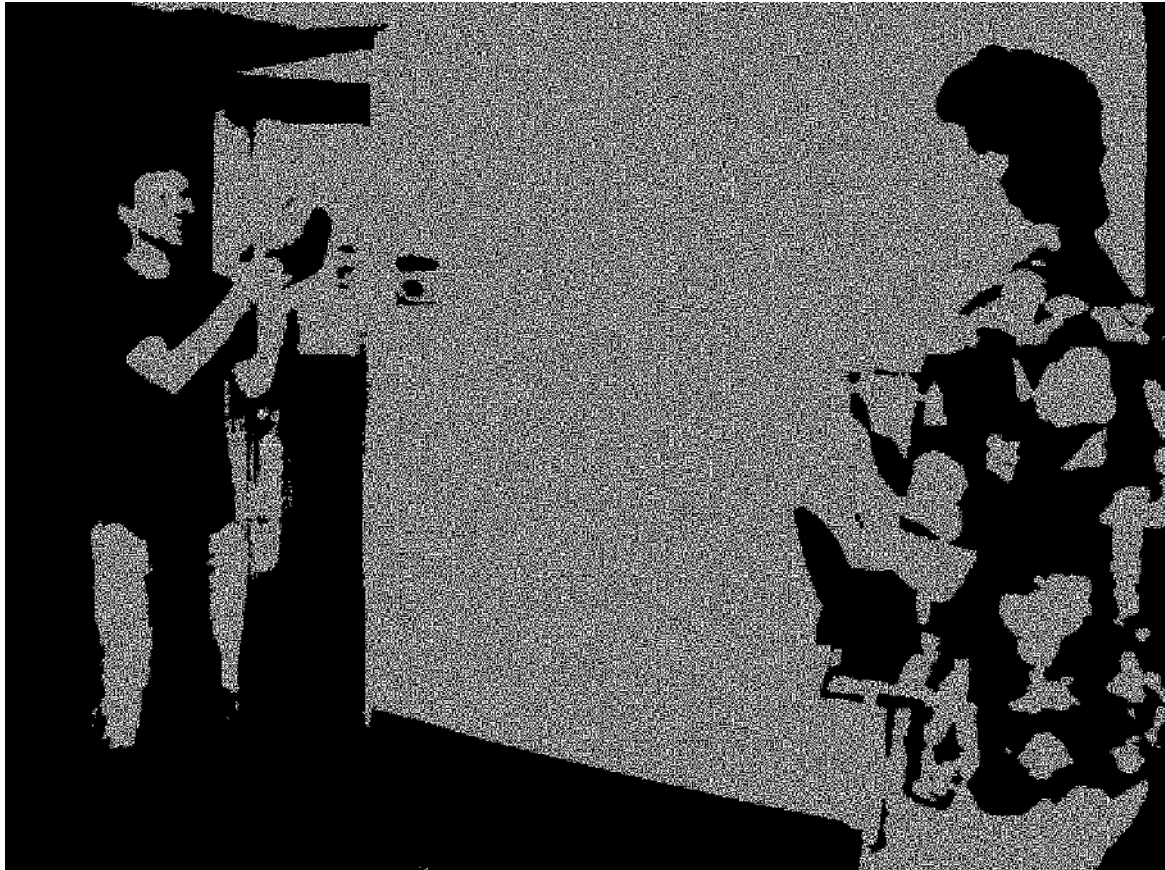
Above is the original image that we will use for the scheme. The dimensions are (640, 480).

Below are the shares after passing through our scheme.



Notice that the dimensions of the image are twice of the dimensions of the original image. This is because we chose $d = 4$ and arranged it as 2×2 block.

Here lies the recovered image. Which is the same dimensions as the share.



Visual Cryptography with coloured images

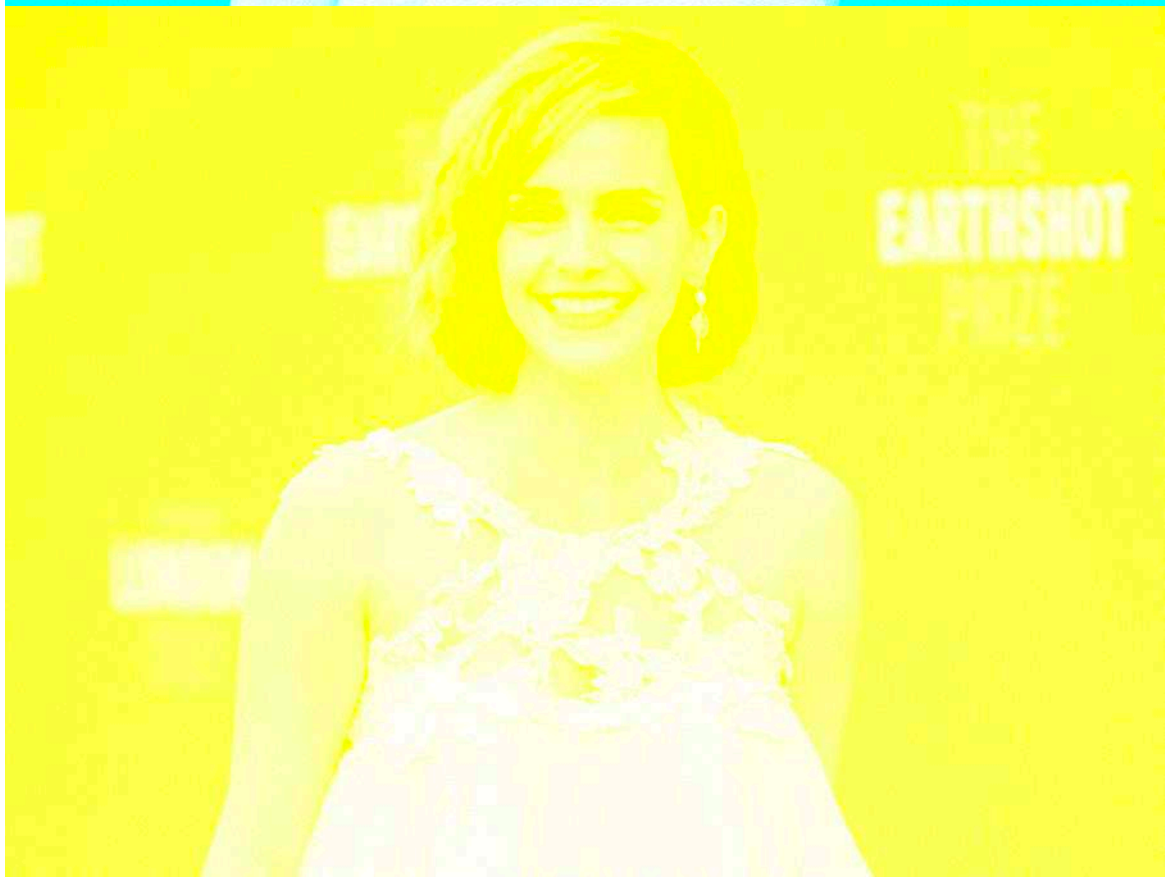
Method 1

The first method we employed had the following scheme. two-level security controlling practice. For example, as long as a manager of a company keeps the black mask of a secret image and gives the rest three shares to his subordinates, the content of the image will remain confidential, even though all his subordinates plot to steal the secret information. Thus, under these circumstances, the black mask share can be regarded as the signature of the manager.

Here is the original image that we plan to share



The method involved decomposing the image into its Cyan, Magenta, Yellow **halftones**, as these are the primary colors used in printing. Halftones are modified versions of the color decomposition of the original image where each pixel is either on or off. There is no in between. Halftones use density to show darkness and lightness.



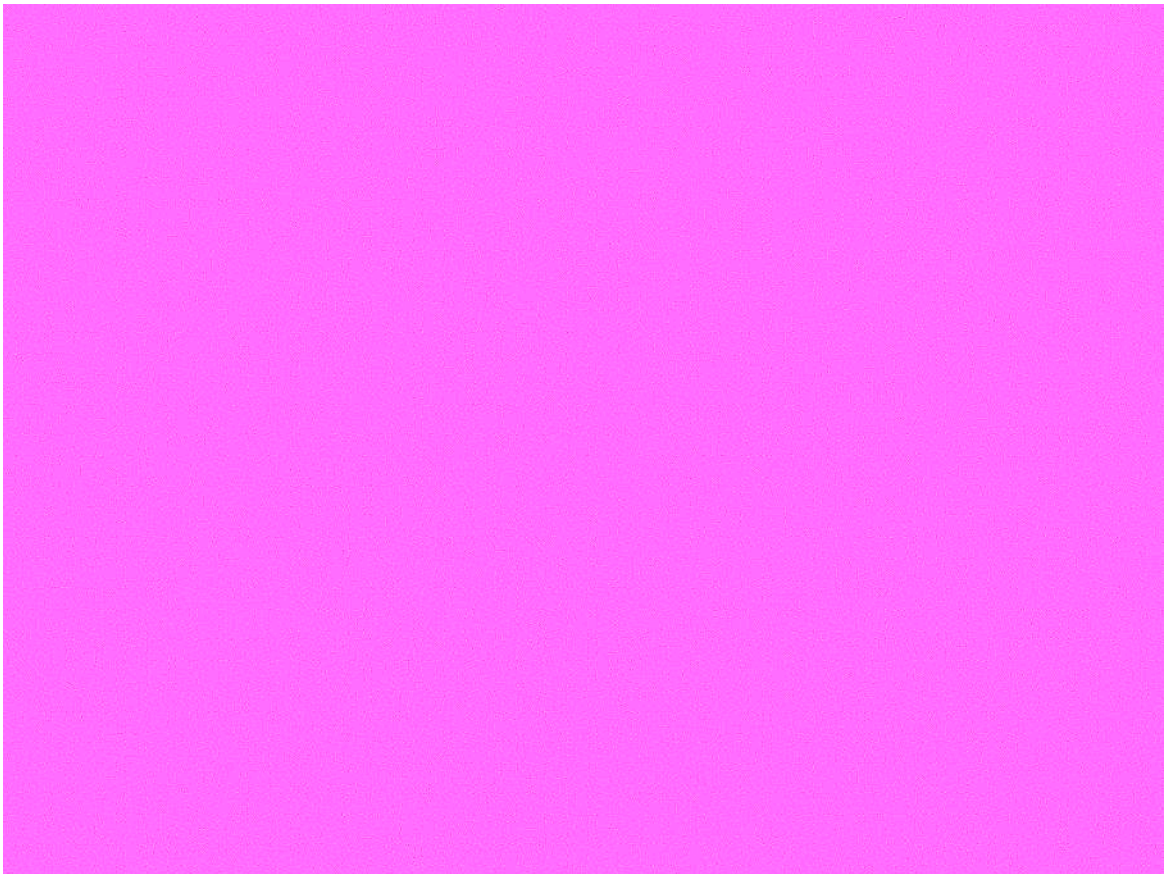
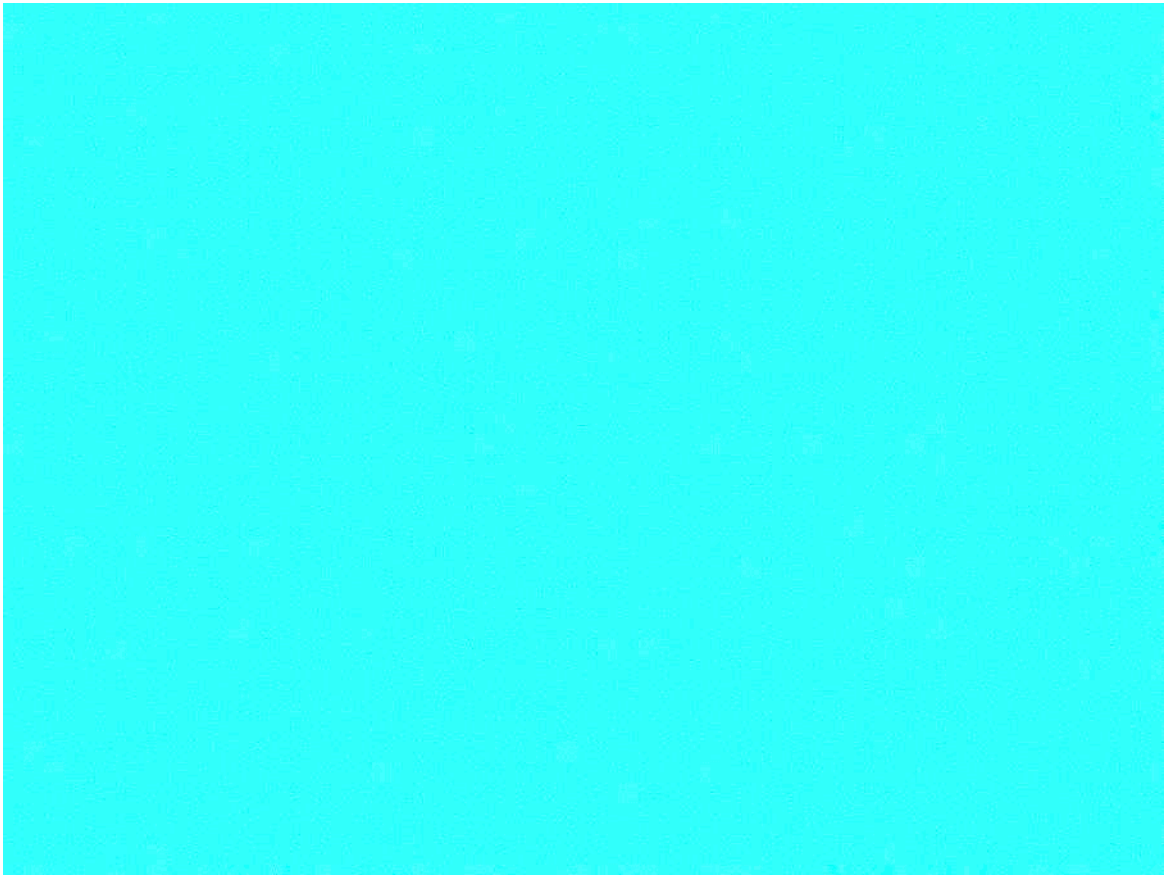


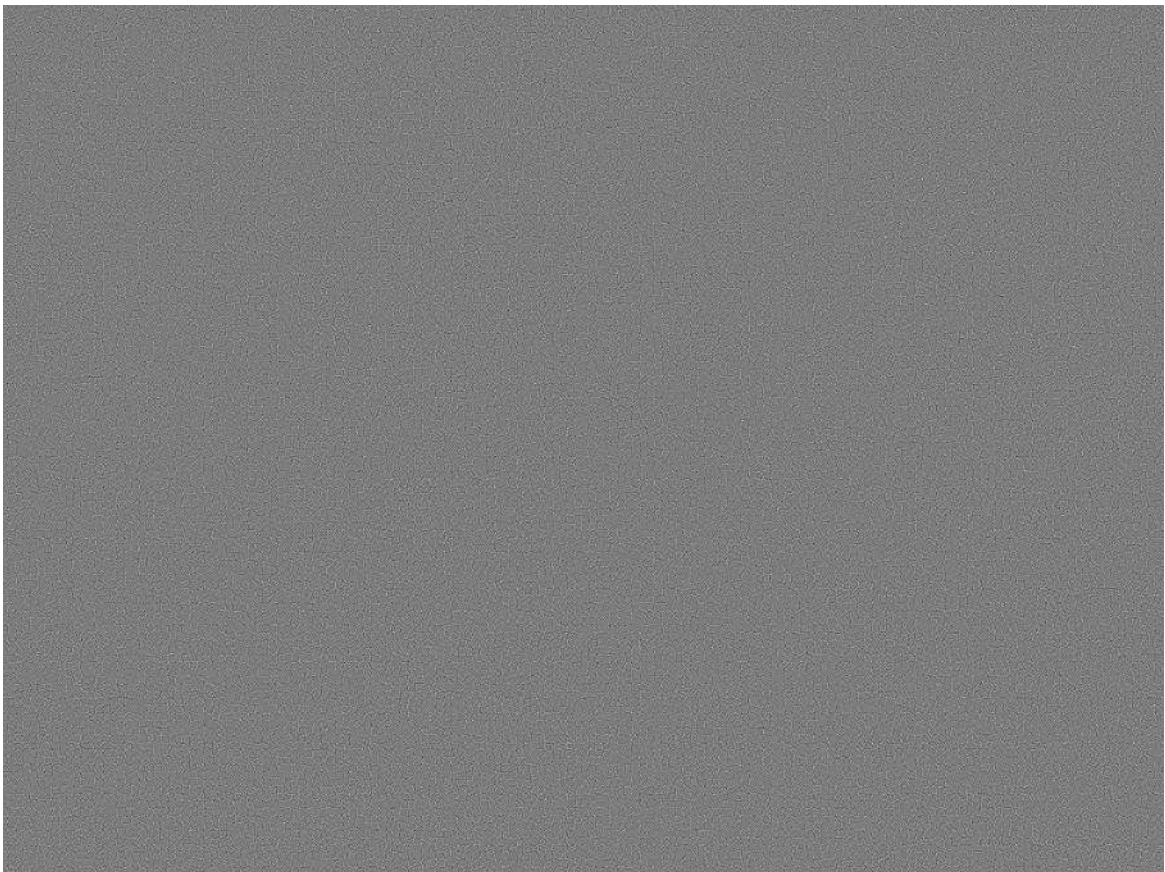
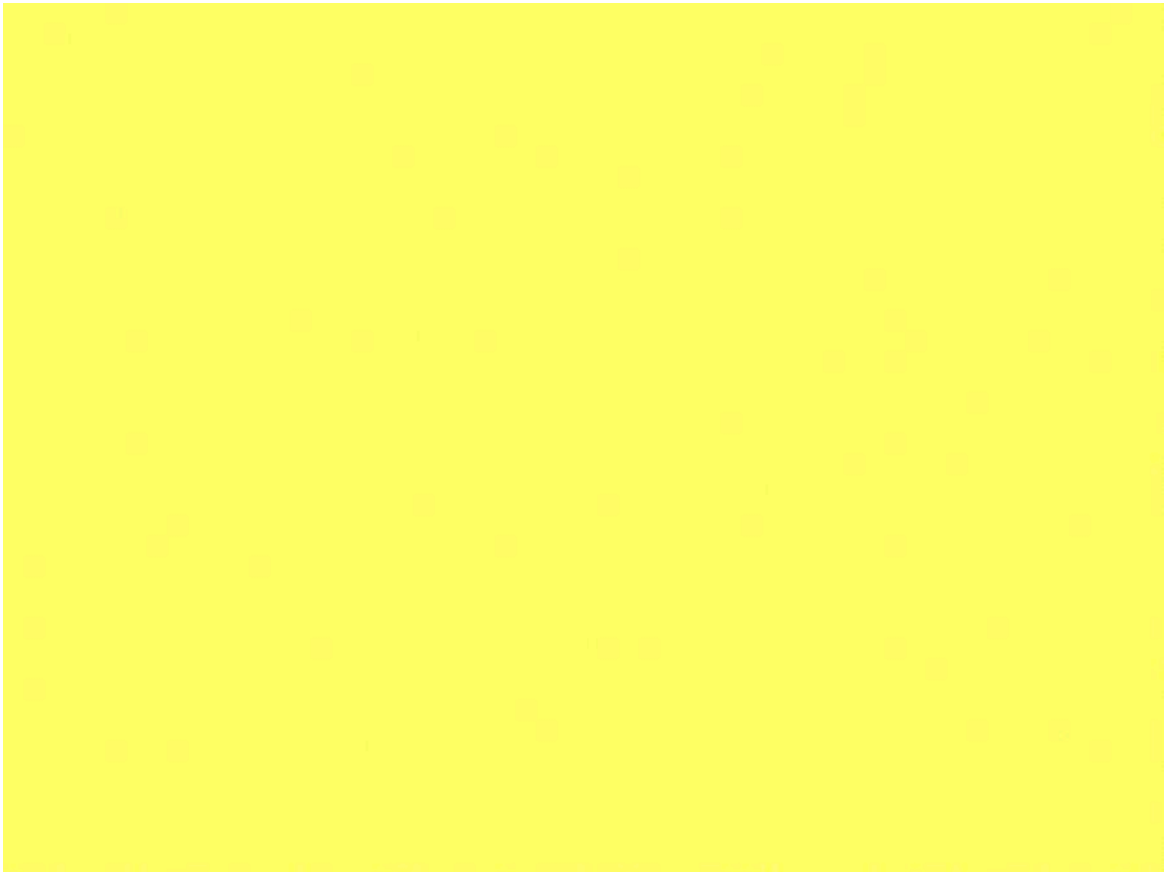
We then proceed to use our algorithm to convert these halftones to shares with a mask. We map each pixel to a 2x2 block and cover half of it with black randomly, and the rest we fill such that value after mixing is same as the original color desired. We used this table to do those calculations.

Mask	Revealed color (C,M,Y)	Share1(C)	Share2(M)	Share3(Y)	Stacked image	Revealed color quantity (C,M,Y)
	(0, 0, 0)					(1/2, 1/2, 1/2)
	(1, 0, 0)					(1, 1/2, 1/2)
	(0, 1, 0)					(1/2, 1, 1/2)
	(0, 0, 1)					(1/2, 1/2, 1)
	(1, 1, 0)					(1, 1, 1/2)
	(0, 1, 1)					(1/2, 1, 1)
	(1, 0, 1)					(1, 1/2, 1)
	(1, 1, 1)					(1, 1, 1)

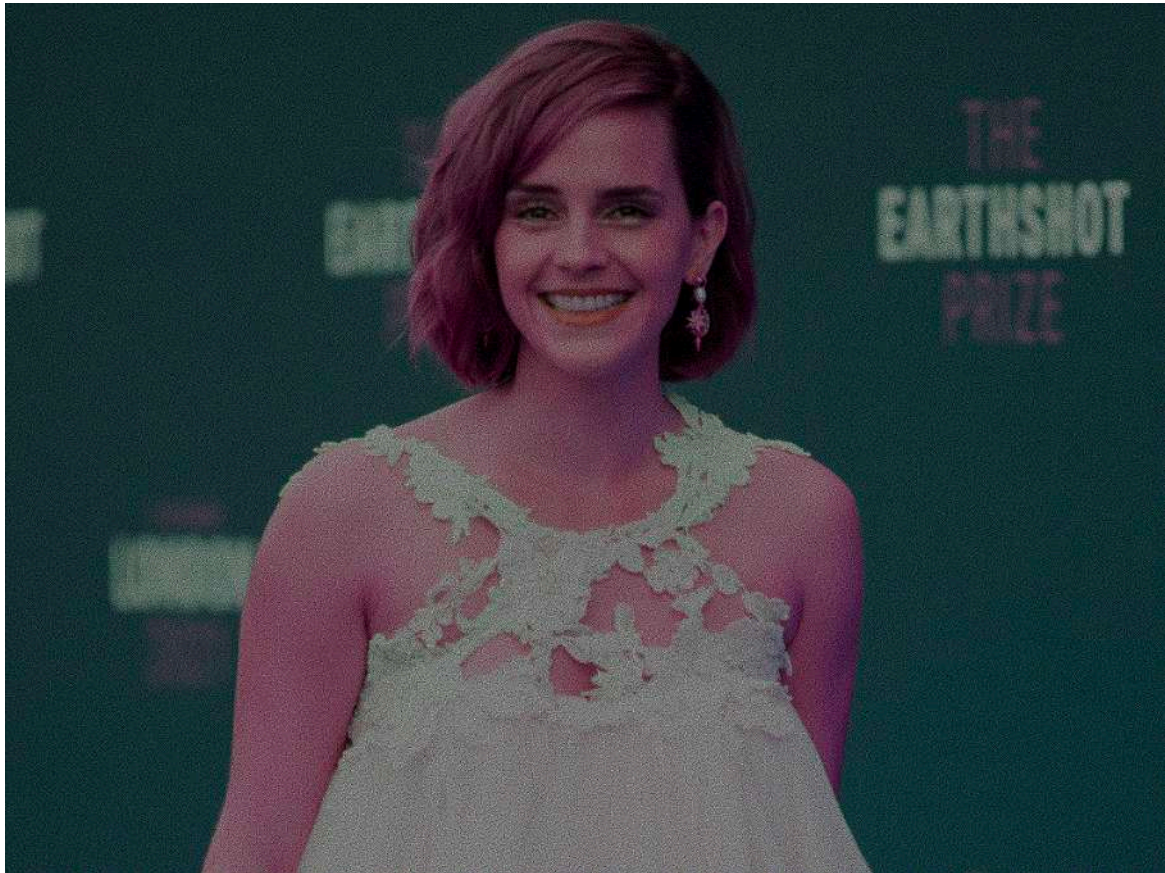
Fig. 8. Scheme 1 of color cryptography.

Here are the shares obtained





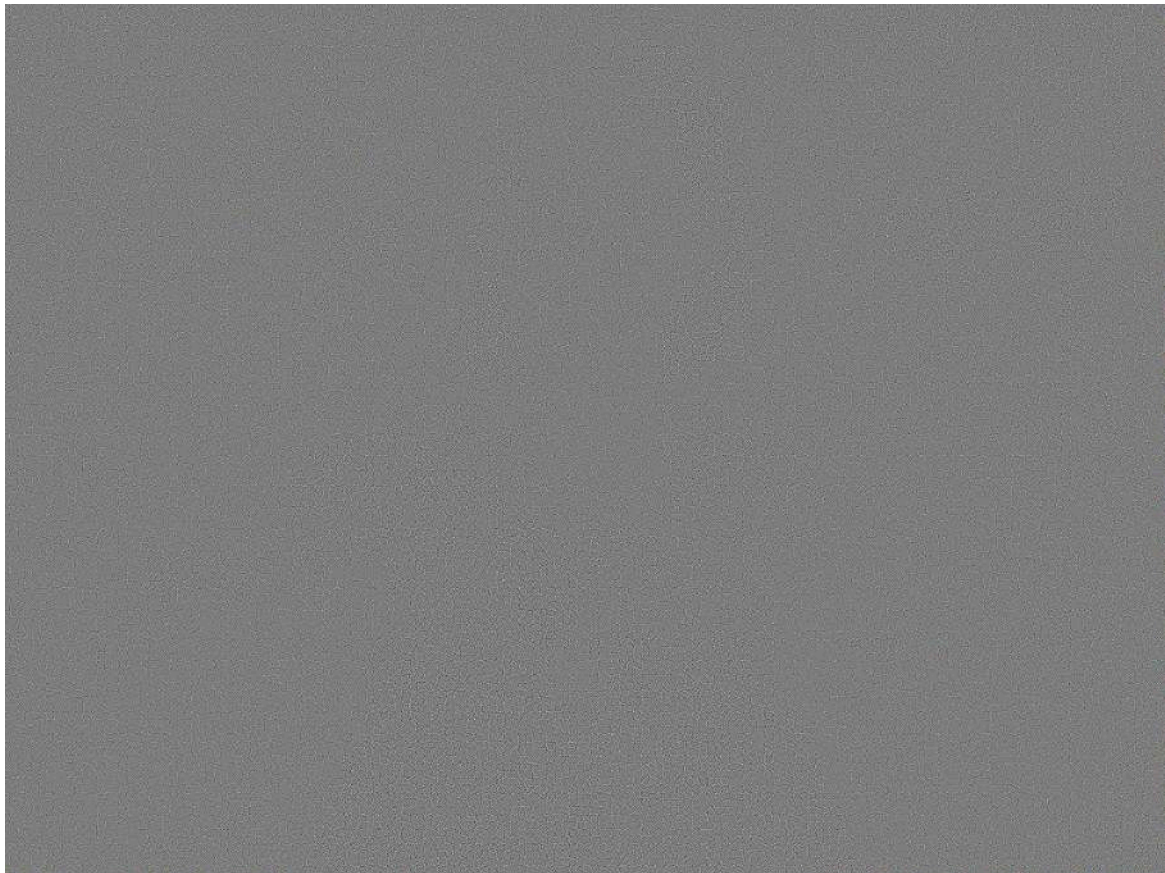
Now the image we recovered after this process was :



This image is at 50% intensity of the original image and 4 times the dimensions, just like the shares.

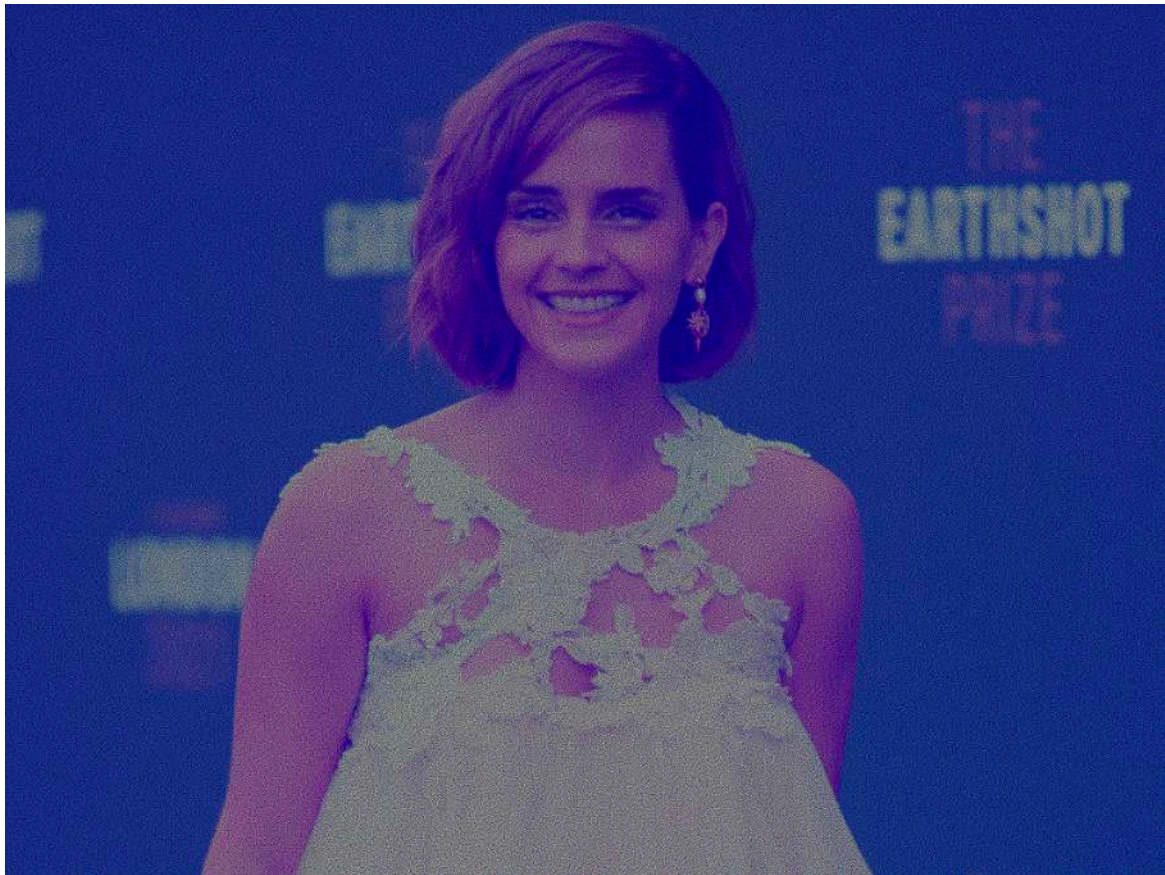
We then did some further analysis to see how much information we are able to get with the mask and without it.

This is the image we got by superimposing the 3 shares



Clearly the workers wouldn't be able to retrieve the image with using just their shares, as shown in the setup.

This is the image we got by superimposing 2 worker shares and 1 mask



Clearly the manager's mask is powerful enough to retrieve most of the data without the need of all 3 workers

This is the image we got by superimposing 1 worker share and 1 mask



Clearly even 1 share is enough for the manager to retrieve the information. Hence this scheme would work great as either a manager's signature or as a 2 level security control.

Method 2

In this method we will create two seemingly similar looking shares unlike the previous method where four shares were created each representing four components of CMYK color scheme.

The setting is as follows -

We have a secret image and we want to create two shares out of it. Each share doesn't reveal any information about the image. Only when both shares are stacked together, the secret image is revealed.

To create the share we first create the three halftone images C, M, and Y as done in previous method. Now, let us focus on how a single pixel from each of three halftones gets translated to a 2x2 block in each of the 2 shares.

We have 3 input pixels, one from each of three halftones.

There are total of 8 possibilities for the input as shown below

































(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)

0 denotes :pixel is OFF

1 denotes :pixel is ON

The 4 pixels (2x2 block) from share-1 comprise of one cyan, one magenta, one yellow, and one white(transparent) pixel which are randomly permuted. and the share-2 is created as per the pixels present in share-1.

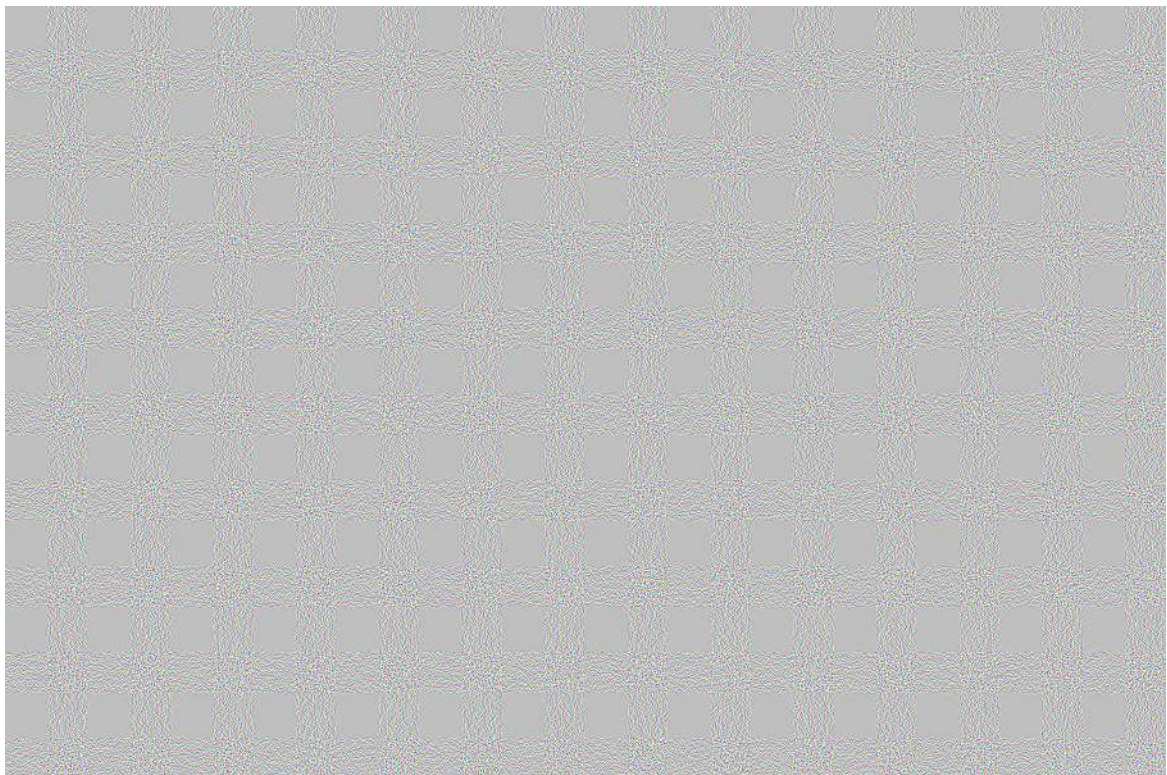
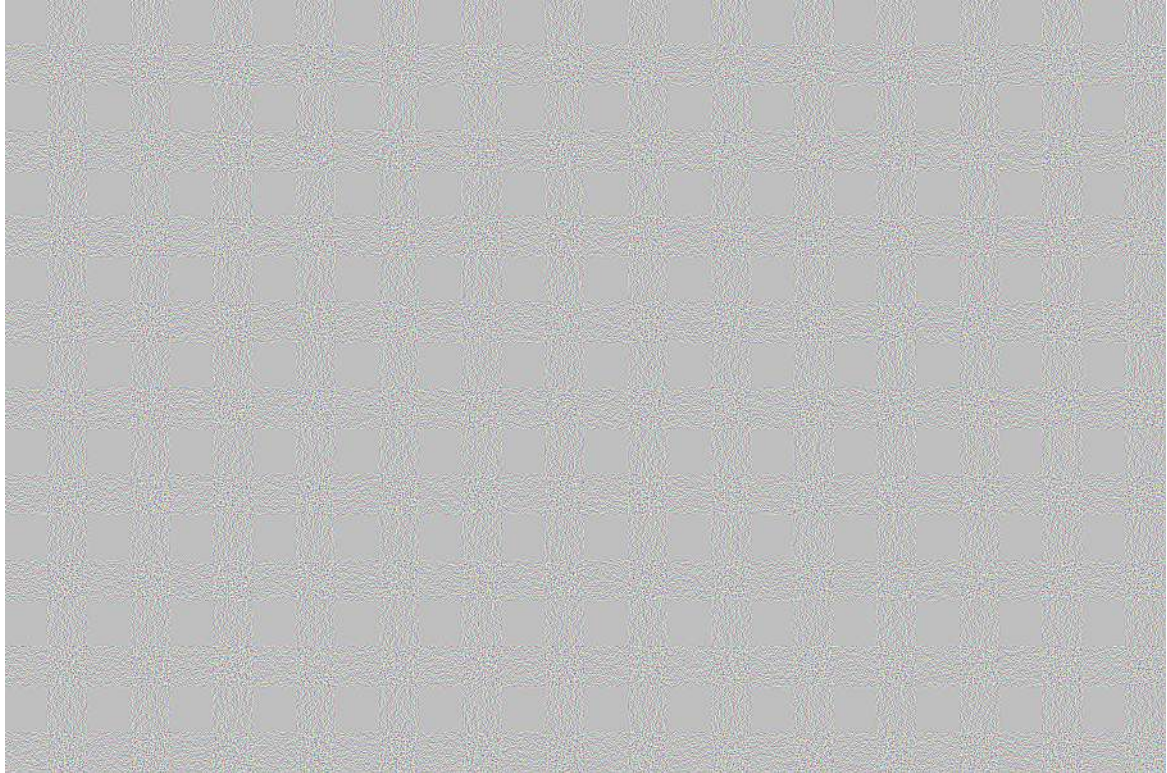
The shares are created as per the table below:

Revealed color (C,M,Y)	Share 1	Share 2	Stacked image	Method	Resultant result	Revealed color quantity (C,M,Y)
(0, 0, 0)				Share 1 and Share 2 with the same permutation		(1/4, 1/4, 1/4)
(1, 0, 0)				Swap the position of cyan and transparent		(1/2, 1/4, 1/4)
(0, 1, 0)				Swap the position of magenta and transparent		(1/4, 1/2, 1/4)
(0, 0, 1)				Swap the position of yellow and transparent		(1/4, 1/4, 1/2)
(1, 1, 0)				Swap the position of cyan and magenta		(1/2, 1/2, 1/4)
(0, 1, 1)				Swap the position of yellow and magenta		(1/4, 1/2, 1/2)
(1, 0, 1)				Swap the position of cyan and yellow		(1/2, 1/4, 1/2)
(1, 1, 1)				Swap two positions in pair		(1/2, 1/2, 1/2)

This is the original image we used



The shares obtained were



The recovered image



And here is the final image that we recovered by superimposing one over the other

The final image has $\frac{1}{4}$ th the intensity of the original image and its size is 4 times that of the original image.

Sidenotes

- The main problem we encounter here is that of addressing the issue of the big size of the image that comes with a larger sharing scheme, as we need more room to add different ways to share each pixel.
- We introduce a third way to address the inconvenient nature of Method 1, which requires the use of four sharing images, as well as the loss of image contrast under Method 2. This method requires only two sharing image and does not sacrifice too much contrast for color visual cryptography.

Firstly we transform the given image in three halftone images C, M, and Y as done before. Next, we generate six temporary shares C1, C2, M1, M2, Y1, and Y2. Each of these sharing images will have two white pixels(transparent) and two color pixels in every 2×2 block i.e. all the color quantities are 50%. The share-i will be created by merging image Ci, Mi, and Yi.

Let us focus on how to create the share C1 and C2 (similar argument will follow for M and Y). Consider a single pixel in halftone image C. There are two possibilities - pixel is ON or OFF.

if pixel is ON, C1 and C2 will have cyan pixel in complementary positions of 2×2 block corresponding to this pixel.

For eg:

C1	C2
white cyan	cyan white
cyan white	white cyan

This will ensure a complete 2×2 block of cyan when shares are stacked together.

if pixel is OFF, C1 and C2 will have cyan pixel in same positions of 2×2 block corresponding to this pixel.

For eg:

C1	C2
white cyan	white cyan
cyan white	white white

This will ensure a transparency in 50% of 2×2 block when shares are stacked together.

Our Codes

- Implementation of Basic Visual Cryptography and Method 1 of Color Visual Cryptography : [link](#)
- Implementation of Method 2 of Color Visual Cryptography : [link](#)

References

1. <https://fardapaper.ir/mohavaha/uploads/2018/12/Fardapaper-A-Comprehensive-Study-of-Visual-Cryptography.pdf>
2. <https://link.springer.com/content/pdf/10.1007/BFb0053419.pdf>
3. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.5077&rep=rep1&type=pdf>
4. <https://stackoverflow.com>
5. <https://pillow.readthedocs.io>