Visual Cryptography

Normally we depend on computers to do computations hefty for human beings to do manually in order to reveal the secret shared through some cryptographic means in a secure manner. But with the technique of Visual cryptography presented in the paper titled *Visual Cryptography* by Moni Naor and Adi Shamir we can rely on the human visual system instead of a computer to “decrypt”. We look at the problem as a k out of k secret sharing problem where a secret image is to be shared with k people and any r<k number of people can not see the secret image and only after stacking k transparencies on top of each other the secret message will be revealed. We implemented this cryptographic system for black and white images but it could be extended to grayscale as suggested in the mentioned paper. The objective here is to generate k transparencies for a secret image with the property that anyone who has r<k transparencies is as good as someone who has no transparencies at all and anyone with k transparencies can simply stack the transparencies over each other and see the secret image without doing any computation.

The basic model for the scheme is that we look at the secret image as a 2-d matrix of pixels each of which are either black or white and in the transparencies we represent each pixel of the original secret image with a rectangular block of pixels which we’d like to be square in shape if possible so that the aspect ratio of the secret image remains same.

We’ll have two set of k $\times$ m boolean(where 0 represents white and 1 represents black) matrices C0 and C1, for replacing a white pixel in each of the k different transparencies with rectangular block of size m pixels(m should preferably be a perfect square number so that the replacement blocks can be square shaped and hence the aspect ratio would be preserved)

Intuitively the scheme needs to satisfy 2 conditions:

1. On laying the k rectangular blocks used to represent any particular pixel ‘p’ in the k transparencies should look blackish if p is black and whiteish if p is white.
2. On laying any r < k rectangular blocks used to represent any particular pixel ‘p’ in the k transparencies should give away no information about whether a white or a black pixel was being shared.

Formally if V represents the bitwise ‘or’ of all the rows of any matrix in C0 or C1 then:

1. For every matrix in C1 the hamming weight H(V) >= d and for every matrix in C0 H(V) <= d - ma (where a is the relative difference in H(V) for any S0 in C0 and any S1 in C1).

2. For any {i1, i2 . . . iq} $\subset$ {1, 2, . . . k}: q < k the two collections of matrices D0 and D1 obtained by restricting matrices of C0 and C1 to rows i1, i2, . . . iq should have the same matrices with same frequencies(*Visual*

Cryptography, Moni Naor and Adi Shamir)

For 2 out of 2 case the following matrices do the job:

C0 = set of all the matrices obtained by permuting the columns of:

0	0	1	1
0	0	1	1

C1 = set of all the matrices obtained by permuting the columns of:

0	0	1	1
1	1	0	0

For a slightly more general k out of k scheme:

We construct C0 and C1 from all the permutations of $k \times 2^{(k-1)}$ matrices S0 and S1 respectively.

Where S0 and S1 are constructed as follows:

even = set of all the even cardinality subsets of the set {1, 2, . . .k}

odd = set of all the odd cardinality subsets of the set {1, 2, . . .k}

Any particular column of S0 will be determined by a unique element 'e' of the set even

For all e in even:

    For all i from 1 to k (S0[i , j] = 1) iff (i in e)

For all o in odd:

    For all i from 1 to k (S1[i , j] = 1) iff (i in o)

After this:

C0 = set of all the matrices obtained by permuting the columns of S0.

C1 = set of all the matrices obtained by permuting the columns of S1.

Here:

d = $2^{(k-1)}$

m = $2^{(k-1)}$