

# FULL RESTORATION OF VISUAL ENCRYPTED COLOR IMAGES

SIMEON PERSSON AND KRISTIAN VÁRNAI

## 1. ABSTRACT

While strictly black and white images have been the basis for visual cryptography, there has been a lack of an easily implemented format for colour images. This paper establishes a simple, yet secure way of implementing visual cryptography with colour, assuming a binary data representation.

## 2. INTRODUCTION

Traditional methods for visual cryptography have been established, are consistent and easily understood. Unfortunately, these methods exist for black and white pictures only, leaving the encryption of colour images wanting. While there are a handful of attempts at bringing colour to visual cryptography, it is still an open field with implementations of varying efficiency. In this paper, we will establish some basic standards for encrypting colour pictures, as well as a simple, yet efficient method for encryption based upon those rules.

While black and white pictures are fairly easy to work with due to their simple nature, colour pictures contain much more information and possibly details that may not be lost in the process. This leads to a need for the static that normally appears in the decryption of black and white pictures to have to be absent[1]. However, a partial reconstruction of the picture (such as having less than all the necessary parts for full restoration) may not hint at what the final image is meant to be.

In addition to the standard of security for the traditional black and white pictures, we set the following points as mandatory for encryption of colour images.

- Full restoration upon decryption.
- No indication as to the original image, whether by eye or any other method, when combining a subset of all available parts.
- Usability for any type of image, whether that image contains a mixture of colours, is black and white or is simply one single colour.
- Destruction of intermediary steps in the encryption process.

Our method of encryption is, due to the process of creating images within visual cryptography and the expanse of computer use, based around the RGB colour model for computers and other, similar devices. This does not prevent implementation of this process with any other model as long as the information is stored as bits.

---

2010 *Mathematics Subject Classification.* 94A60.

*Key words and phrases.* visual cryptography, color images, one-time pad.

### 3. ENCRYPTION

The encryption is fairly straightforward and can be easily understood as well as implemented. Although it is easy to handle, it is efficient, provides the necessary security and fulfills every point previously stated. For the process, the thought was to work on the bitwise level that represents the colours themselves; in this case, the RGB values are used. As each pixel is processed, two random values are generated; the first one is compared to the RGB value of the current pixel and it is then separated into two values: an RGB value with the bits that were set in both the original as well as the first random value, and another one with the set bits left over from the original.

Next, the second random value is compared to the two new results from the previous step. If both values are not set while the bit at the same position in the random value is, those bits for the values from the previous step are set. These steps are repeated for every pixel and then the encryption is finished. Decryption is easily done via a bitwise XOR of the RGB values of the two resulting pictures and we effectively have a One-Time Pad implementation on the colour values of an image. The random values from the process of encryption are discarded alongside any other values we might have produced.

The encryption algorithm will “split” the original image into two so called *shadow images*. Let  $p$  denote one pixel in the original image, and let  $s_1$  and  $s_2$  denote the corresponding pixels in the shadow images, respectively. Here  $p$ ,  $s_1$ , and  $s_2$  are vectors, representing the channels used, e.g. red, green and blue in the RGB colour model. The calculations during the encryption are carried out both bitwise and channel-wise.

---

**Algorithm 1** Visual encryption

---

```

1: for all pixel  $p$  in the original image do
2:    $r_1 \leftarrow \text{rand}$ 
3:    $r_2 \leftarrow \text{rand}$ 
4:    $s_1 \leftarrow (p \wedge r_1) \vee (\neg p \wedge r_2)$             $\triangleright$  Corresponding pixel in shadow 1
5:    $s_2 \leftarrow p \oplus s_1$                           $\triangleright$  Corresponding pixel in shadow 2
6: end for
```

---

We illustrate the encryption with an example. Assume that we want to encrypt the image in Figure 1. Then if we apply the algorithm, it may result in the two shadow images in Figure 2—remember that the algorithm is probabilistic. To restore the original image, we compute  $s_1 \oplus s_2$  for each pixel. The result will give us back the original image, without any loss of quality, that is, Figure 1.

### 4. VALIDATION OF THE SECURITY

As with both visual cryptography and the one time pad, this method offers complete security as there is neither a repetition to be found, nor is a brute force attack possible as every possible result within the picture’s resolution will show up. This method does not only fulfill the standards we previously set, it even leaves the resolution of the original image intact. Of course, it should be noted that for this encryption to work to its full potential, the results must be saved as a lossless image type. In the event that one would wish to separate the original into more



FIGURE 1. Original image

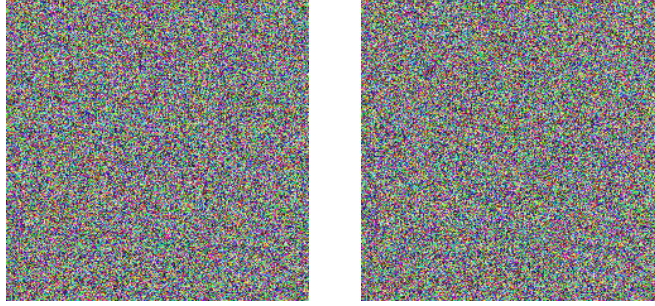


FIGURE 2. Shadow images

than two pictures, reapplying this process to the results until a satisfactory amount is reached is all one needs to do.

It could be argued that all this encryption would need is random values generated and applied with a bitwise XOR to the original image, leaving the randomly generated sequence as one of the resulting images and the result of the bitwise XOR as the other. While this could be done, let us take a look at the absolute worst case scenario, disregarding the possibilities that an attacker knows the original or has access to all parts of the picture. The scenario in mind would be in the highly unlikely event that an attacker would have an intimate enough knowledge of the encryption process to know exactly how it is implemented as well as knowing exactly which random values were generated.

If the method of encryption would be nothing more than the simplified version suggested, then having the result of the bitwise XOR in your possession would be enough to get the original as being able to predict the exact pseudo-random values would mean that you effectively have the key. This is not the case for the method we propose as depending on the random values and how they interact with the original data, only some bits could possibly be decided, as shown in table 1. There are four possible cases for the pair  $(r_1, r_2)$ . Each case can be divided into two subcases, where each corresponds to the possible value  $q$  of  $p$ . Without loss of generality, we consider the cases on bit-level.

As seen, even in the case of knowing the generated pseudo-random values, half the possible combinations lack a value, thus mitigating such an attack. The method of encryption presented here could also be applied to the one time pad due to the

Case	$r_1$	$r_2$	$q$	$p$
1	0	0	0	–
2	0	0	1	1
3	0	1	0	1
4	0	1	1	–
5	1	0	0	–
6	1	0	1	1
7	1	1	0	1
8	1	1	1	–

TABLE 1.

likeness of them, so providing the extra bit of security by mitigating this most unlikely of scenarios. The reasoning for the received original value presented in the above table is shown with simple boolean algebra. The equations used are the ones presented in the pseudocode, also presented here for ease of following. Keep in mind that we do not know which one of the values we have and thus not what the other one could be.

**Case 1 and 2.** Here is  $s_1 = 0$  and  $s_2 = p \oplus 0$ . While we know for sure that at least one of the values must be 0, the other could be either possible value. This means that if our available result ( $q$ ) is 0, we have no way of knowing which value we have in our possession, which in turn means that  $s_1$  and  $s_2$  are equal in terms of possibility. Thus, since the original value could be either for  $s_1$ , it remains undetermined. If, on the other hand, our available value is 1, then we can logically conclude that it is  $s_2$  that we are dealing with and we only have to complete the following equation:  $1 = p \oplus 0$  and therefore  $p = 1$ .

**Case 3 and 4.** Here is  $s_1 = \neg p$  and  $s_2 = 1$ . As per the same logic as in the previous one: if the available result is the same as the one we know must exist (in this case one), then we can not tell which equation it is we have. If it is the opposing value, then we can in this case rule out  $s_2$  and calculate the equation:  $0 = \neg p$  and hence  $p = 1$ .

**Case 5 and 6.** Here is  $s_1 = p$  and  $s_2 = 0$ . Following the previous logical steps gives us an unknown for the available result of 0, but the following result for 1 is  $p = 1$ .

**Case 7 and 8.** Here is  $s_1 = 1$  and  $s_2 = p \oplus 1$ . Once again we can not say for sure which case is before us if our available result is the same as the one we know with certainty to exist, but in the case of a differing result, we know the following:  $0 = p \oplus 1$  and  $p = 1$ .

## 5. ACKNOWLEDGEMENTS

We would like to thank Robert Nyqvist for:

- Supplying the necessary information
- Helping with the writing of this paper

## REFERENCES

- [1] Duo Jin, Wei-Qi Yan, Mohan S. Kankanhalli  
*Progressive Color Visual Cryptography*  
[http://www.comp.nus.edu.sg/~mohan/papers/prog\\_vc\\_jei.pdf](http://www.comp.nus.edu.sg/~mohan/papers/prog_vc_jei.pdf)

*Current address:* Blekinge Institute of Technology, Undergraduate Student

*Current address:* Blekinge Institute of Technology, Undergraduate Student