ENCRYPT
PRINT
DECRYPT

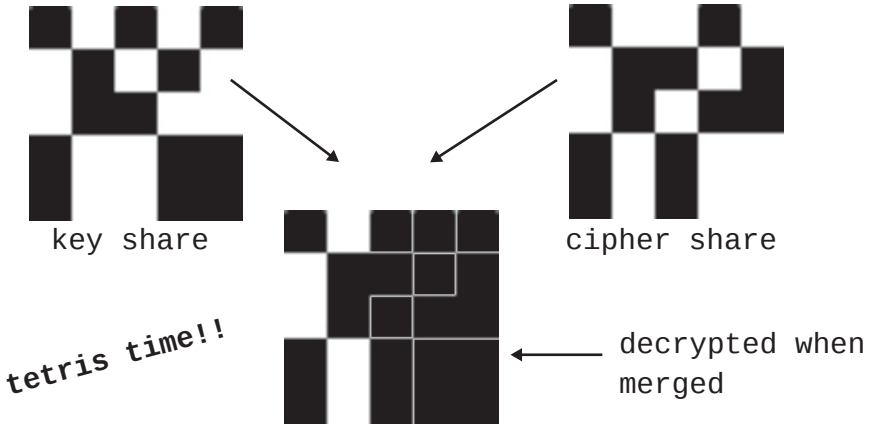
A ZINE ABOUT SECRET WRITING WITH VISUAL CRYPTOGRAPHY

INDEX

What is Visual Cryptography

Visual cryptography (aka VC) is a technique which allows visual information such as pictures, or text on an image, to be encrypted by splitting the information in two or more shares and applying color/shade swapping. It can be decrypted "visually" by superimposing the shares of the image without requiring a computer. Examination of one share discloses no information about the encrypted message.

A use case example: As part of a secret exchange, a group sends to their members a transparency (key) by post. Same group sends via email or displays on their website the cipher (encrypted message). Members overlay the key on the cipher, read the secret key and communication is established.
The concept was introduced at EUROCRYPT'94 conference on cryptography, still happening today at eurocrypt.iacr.org

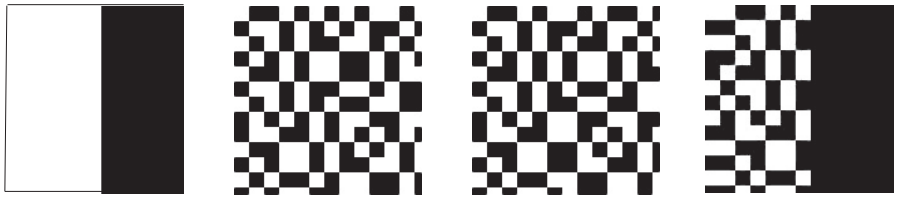


How does it work?

Let's say we want to encrypt the image with two shares, meaning that it will consist of one key and the cipher. The key is a random black&white generated image of the same size as our message-image.

If "P" is black in the message, then the subpixels of the key and the cipher (one white, the other black) compliment each other so when they are superimposed the whole pixel is black again.
If "P" is white, then the subpixels in the key and in the cipher have the same shade in the same position, so when they are overlayed they give gray (read as white). Since all the subpixels in the key are colored randomly, subsequently so are they in the cipher, so no information can be gained by looking at any one share.

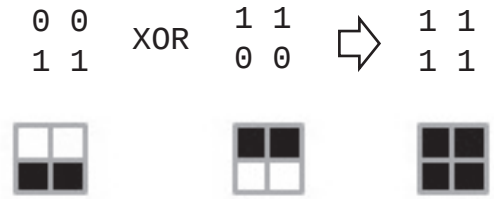
However there will be a 50% loss of contrast in the reconstructed image due to the white pixel becoming gray, but it should still be visible.



Algorithms for VC

The logic behind this pixel color swap scheme is XOR (aka exclusive OR or exclusive disjunction, math symbol *V*), which outputs true only when inputs differ; one is true, the other is false, but not both. https://en.m.wikipedia.org/wiki/Exclusive_or

```
1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 = 1
0 XOR 0 = 0
```



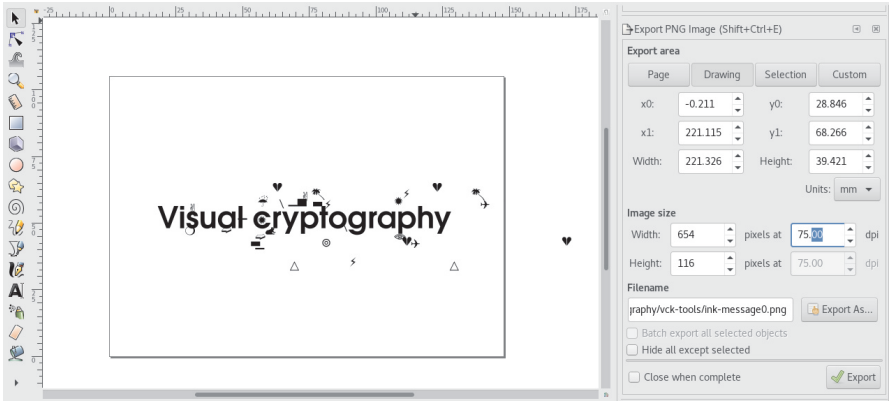
To securely encrypt a message with XOR only, you need a key that is as long as the message, which is why in VC the random generated key image has the same size as the message image. If such a key is completely random, best generated with external input and hardware instead of a software's pseudorandom algorithm, and you never reuse it, the encryption known as "one-time pad" is unbreakable.

Fan fact 1:XOR is its own inverse. So, if m is a message and k is a key, then k can be used to both encrypt the message e=m*V*k and decrypt m=e*V*k
Fan fact 2: XOR-ing is such a fast computation that on some computer architectures, it is more efficient to store a zero in a memory register by XOR-ing the register with itself (bits XOR-ed with themselves are always zero) instead of storing the value zero.

Do the secret writing

Start with the message you want to encrypt.
Here we will do an example using text and we will go through three ways to convert text to image.
Tools to install, if not yet have, in your computer: gimp and either imagemagick, or inkscape, or scribus.

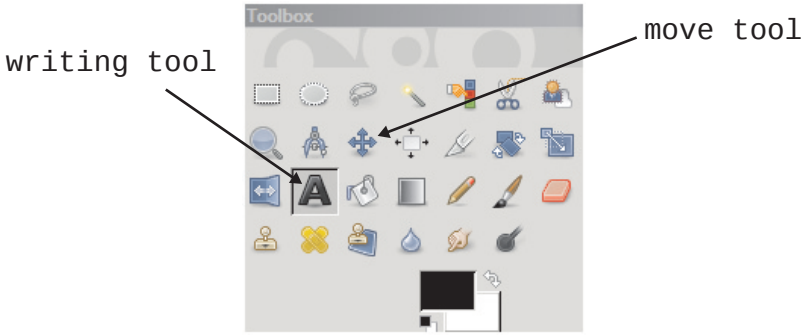
1. with inskscape (or with scribus is similar)
Open the application, under File > Document Properties, choose a A5 or A6 document (landscape or portrait), and click on background and choose white. In the topbar click View > Color display mode > Grayscale. Then click on Text and choose a font, size about 30 and bold, click Apply. From the left side-bar click on the icon A to write your text. Then click File > Export PNG image, and in the Drawing tab > Image size enter pixels at 75.00 dpi. Give a filepath and click Export.



Convert message to image

2. with imagemagick (IM)
Open a terminal and run:
convert -background white -fill black
-font <path-to-fonts>/Cimatics -pointsize 32
-gravity center label:'My secret!' message.png
NOTE 1: for more examples see IM docs:
https://www.imagemagick.org/Usage/text/#label
NOTE 2: find fonts in linux/macos system with
terminal command "whereis fonts"

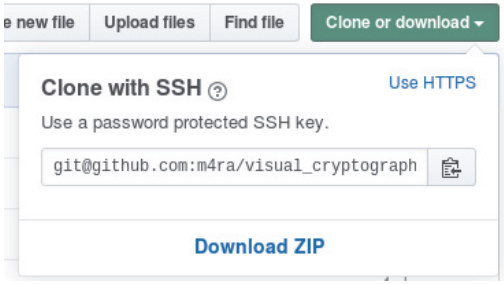
3. with gimp
Open application, go to File > New > Template > A4,
choose portrait or landscape under the Image Size.
Click on Advanced Options > X,Y resolution 75.00,
Color space: Grayscale.
On the new file, from the top-bar click Tools >
Toolbox. In the toolbox click on the font icon, and
enter in the dialogue box the font specifics. Write
the text in the canvas. Choose the move icon from
toolbox to place the text where is desirable. Go to
File > Export as, and choose png image type.



Get the python tools

Now that you have your message in an image format,
let's install the python scripts. You can git
clone the repo
https://github.com/m4ra/visual_cryptography

If you don't have git, then click and download the
files as a zip archive.



Next, you need to install the Python Image Library
version >= 2.0.0
from terminal run:
"sudo pip install Pillow" ot "easy_install Pillow"
for more install options see:
pillow.readthedocs.io/en/3.1.x/installation.html

Install also Tkint, ehich creates window interfaces
during encryption process to display the image layers.
for python 2.7.x, from terminal run:
sudo apt-get install python-tk
for python 3.x.x, from terminal run:
sudo apt-get install python3-tk
see more at: https://tkdocs.com/tutorial/install.html

Encrypt

OK, now we have all our tools let's create the image shares by generating a random image key and a cipher image, and this is how the message will be encrypted.

Go to the installed folder
visual cryptography > vck-tools.
From terminal check your system's defaut python version with "python -V"

For python 2.7.x run:
python vck-split-mono.py message.png
⇧
(here goes the filename of your message-image made before with gimp, imagemagick or inkscape)

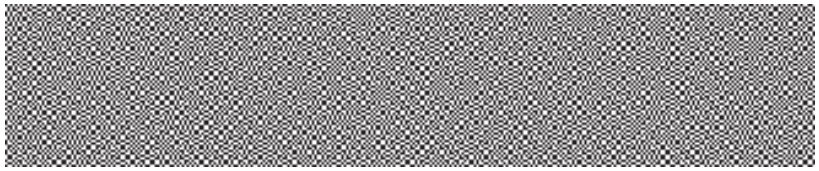
For python 3.x.x run:
python vck-split-mono-py3.py message.png

This produces the key and the cipher images, in this example "message_1.png" and "message_2.png" respectively.

Print and decrypt

You can print both shares (key and cipher) in transparent sheets, or one share in white paper and the other in transparent. Following the steps before, the images to print are:
<filename_of_original>_1.png
<filename_of_original>_2.png
printer options: Scale 100%
When overlaying the shares, images must be aligned precisely so that their borders match.

message_1.png



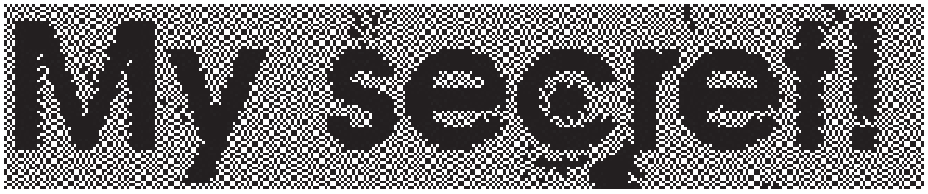
message_2.png

You can also generate a barcode with your secret message, and skip the text-to-image conversion. Others would have to decrypt the barcode and perform barcode scanning to read your message. Here is a free online generator:
http://www.barcodegenerator.online/

Do it on your screen

During the image encryption with python, there is also a "result.tif" file created, which is the decrypted image. You can see the result as if you would by overlaying the printed versions of key and cipher.

Another way is to open the key and the cipher images with gimp. Make sure key image has an alpha transparency (Layer > Transparency > Add Alpha channel), then select all white pixels with Shift+o, and click Edit > Clear. Select the image, copy with Ctrl+C and paste it on the cipher image, and tada:



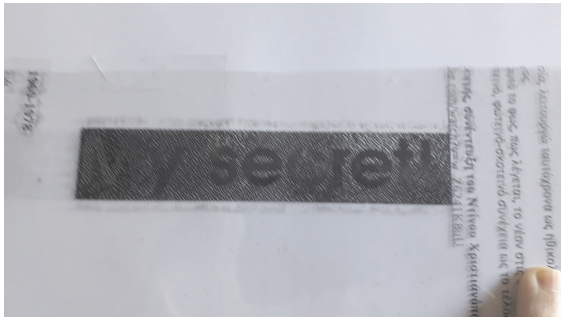
result with text-to-image conversion with imagemagick

and result with using inksacpe

Share your secret

You can share your key by email to others or post the printed-on-transparency key by snail-mail, and then uploading the key on a website.

Since this zine was created as workshop material for the festival "art meets radical openess", participants are invited to upload their expirements at the servus gitlab > Enrypt-Print-Decrypt > Encrypted_Messages. The decrypted results, either photos of prints or screens versions, can be uploaded under folder Decrypted_Messages. If you have any blocks during the process ping me (contact details in the colophon).



More on secret writing

How logical operations such as XOR, OR, NOT, AND work:
https://en.m.wikipedia.org/wiki/Bitwise_operation

What is an uncrackable one-time-pad:
https://en.wikipedia.org/wiki/One-time_pad

Slideshow about VC:
https://www.slideshare.net/AneeshGKumar/visual-cryptography-70058247

Art and design:
https://kai.jauslin.biz/other/visual-cryptography/

https://holesinsmoke.hotglue.me/

https://mara.multiplace.org/rhythmic_translator/

Texts about encoding, decoding writing:
http://avant.org/project/math-and-mysticism/

Writing, Medium, Machine
http://openhumanitiespress.org/books/download/Pryor-Trotter_2016_Writing-Medium-Machine.pdf

The Code book
https://monoskop.org/log/?p=871

Colophon

this zine was made to serve as workshop material for
AMRO 2020
radical-openess.org

icons are hand drawn
layout design with scribus
pictures with gimp imagemagick and inkscape

the font for text is liberation mono

the font for the cover
index and this colophon is Cimatics by
OSP foundry
http://osp.kitchen/foundry/

for zine inquires or other info
email mara@multiplace.org
mastodon mara@systerserver.town