Problem 3. Mixed hashes

Dung Le Quoc

October 22, 2023

By analyzing sample (mikki.ppm) I found that size of image equal $w \cdot h \cdot 3$, where w is width of image and h is height.

Let w_i be the width of i-th image, and h_i be the height of i-th image.

The ciphertext of i-th image will have size equal to $S_i = w_i \cdot h_i \cdot 3 + pd_i$, where pd_i is padding in ECB mode.

Notice that image should be nearly square, which means $w_i \approx h_i$.

So for $i = 1, 2, \ldots, 8$ I calculate $\sqrt{S_i}$ and take the bound for width and height, for example, size is integer in range $[300, 800]$.

For each image, I bruteforce w_i, h_i in this bound, until I found the hash of header **P6 X Y 255** appear in hashes list.

Notice that substraction $S_i - w_i \cdot h_i \cdot 3$ should be less than 8 because of padding of PRESENT and ECB mode.

In the result, I receive ciphertext ct_i and its coressponding hash h_j:

ct_1	h_3
ct_2	h_7
ct_3	h_5
ct_4	h_4
ct_5	h_1
ct_6	h_6
ct_7	h_8
ct_8	h_2

The key for PRESENT cipher is the same for all images, so I guess it is "**P6 X Y 255**" (10 symbols).

The hash comes with size (width and height) of image. Using it I and decrypt and recover all images.

The plaintext is "♡Loveyou" (I'm not sure about uppercase and lowercase).

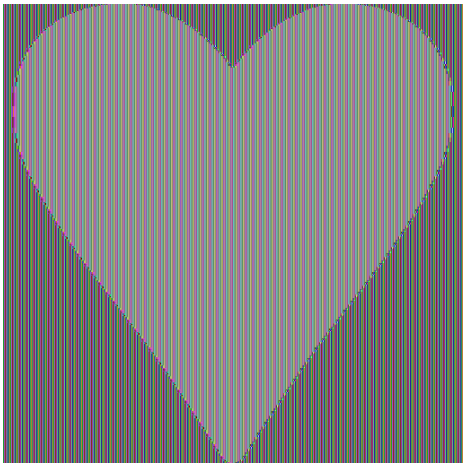

Figure 1: File1.ppm

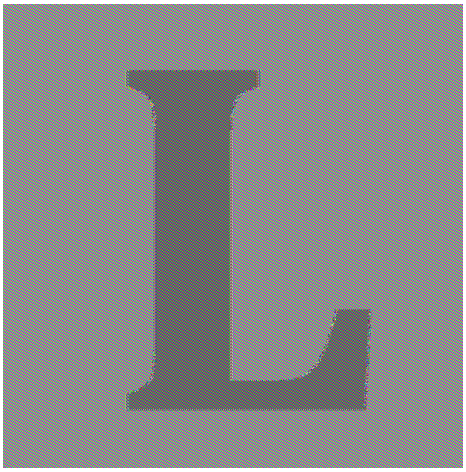

Figure 2: File2.ppm

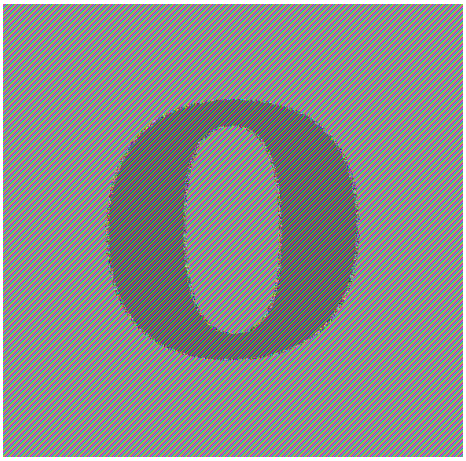

Figure 3: File3.ppm

Figure 4: File4.ppm

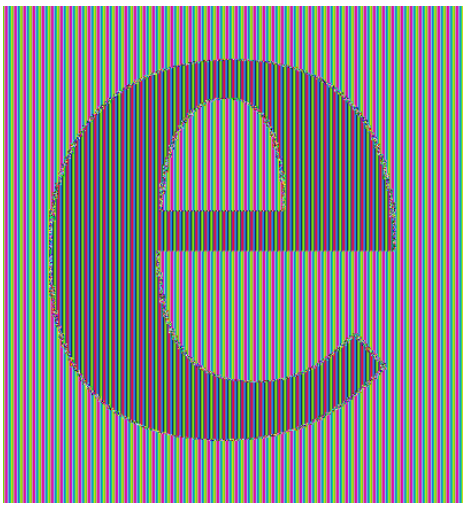

Figure 5: File5.ppm

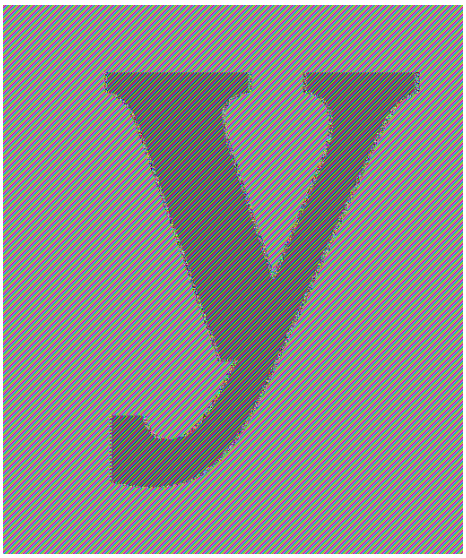

Figure 6: File6.ppm

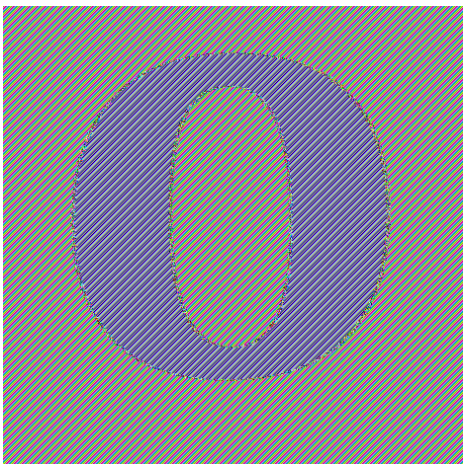

Figure 7: File7.ppm

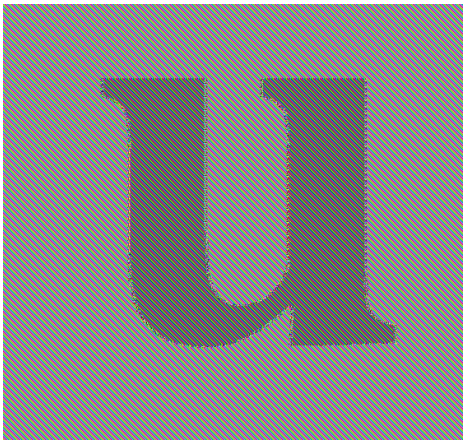

Figure 8: File8.ppm