# Lab 4: CRYPTO

Crypto1 –encode image.ppm

- Read ppm image into 2D array
- Set pixel index list (even pixels)
- Encode text from standard input one character at a time: place one text bit in one RGB LSB bit
- Write to image_wmsg.ppm

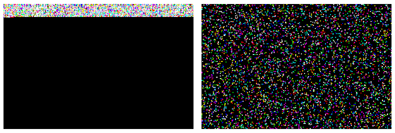
Crypto1 –decode image_wmsg.ppm

- Read ppm image into 2D array
- Set pixel index list (as above)
- Extract RGB LSB bits from pixels in pixel index list: combine into characters and write to stdout.

Crypto2 ... [–seed=N] ...

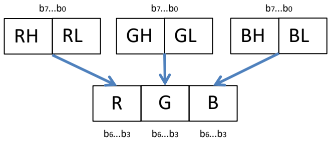
- Permute pixel index list using histogram of 12-bit integers formed from extracted RGB bits.
- Combine two 12-bit random numbers into a 24-bit number which is then used to carry out permutation of pixel index list.
- Optional command line seed

Illustration of encoding patterns

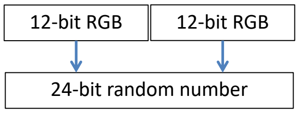


Crypto1 Crypto2

Create 12-bit number from RGB



Create 24-bit number from two random 12-bit numbers



Above used by Crypto2 and Crypto3

Crypto3 [–key="text"] ...

- Optional text key used for XOR-based encryption and decryption.
- NOTE: c=XOR(XOR(c,k)) for any k.

Crypto3 random.ppm image.ppm

- First image used to shape the random number generator
- Second image used to embed text