Day 31, Cryptology and Steganography

Lock Box Example

Cryptology is the study of codes, both creating and solving them.

Cryptography is the art of creating codes, or encryption.

Cryptanalysis is the art of breaking codes, or decryption (https://militaryembedded.com/comms/encryption/cryptology-cryptography-and-cryptanalysis#:~:text=Cryptology%20is%20the%20study%20of,the%20art%20of%20creating%20codes).

So, Cryptology = Cryptography + Cryptanalysis.

For the codes to be useful, they should be relatively easy to encrypt messages, but very difficult to decrypt for those the messages are NOT intended for (and of course relatively easy to decrypt for the those the message IS intended for). They should be ***invertible***.

Many of the techniques we will discuss rely on modular arithmetic. Consider the following: *When **N** items in a basket are removed 5 at a time, ultimately none remain. However, if these **N** items are removed 2 at a time, then 1 remains. If removed 4 at a time, then 3 remain. What is the smallest possible value of **N**?*

First, write this scenario mathematically, using appropriate symbols:

N % 5 = 0	→	5	10	15	20	25	30	35	40	…
N % 2 = 1	→	5	~~10~~	15	~~20~~	25	~~30~~	35	~~40~~	…
N % 4 = 3	→	~~5~~	~~10~~	***15***	~~20~~	25	~~30~~	35	~~40~~	…

What would be the next smallest value of ***N*** which satisfies the above conditions? Can you explain this?

For another example of modular arithmetic, let's consider the Universal Product Codes (UPC) we are so familiar with when scanning items we are purchasing. The most widely used barcode in the US retail industry is the UPC-A or UPC-12, an example of which is shown below (https://www.createbarcodes.com/create-upc-barcode).

The check digit is calculated using a formula based on modular arithmetic. You take all the digits in odd positions (the positions start at 1), sum them, and multiply that sum by 3. Add to this the digits in the even positions, except for the last digit. For this sum, "mod" it with 10. If it is not 0, subtract this from 10. That should be the check digit.

Following this algorithm on the example above:

$$3 * (0 + 2 + 4 + 6 + 8 + 0) = 3 * (20) = 60$$
$$1 * (1 + 3 + 5 + 7 + 9) = 25$$

→ $(60 + 25) \% 10 = 85 \% 10 = 5$ → $10 - 5 = \mathbf{5}$

For more on check codes, you can read: https://www.barcode.graphics/gtin-12/?srsltid=AfmBOooIJ_6D-cCpc16BtfgNgbOl-icmWxIIbToLOU6fmG0vCKDRWSuX

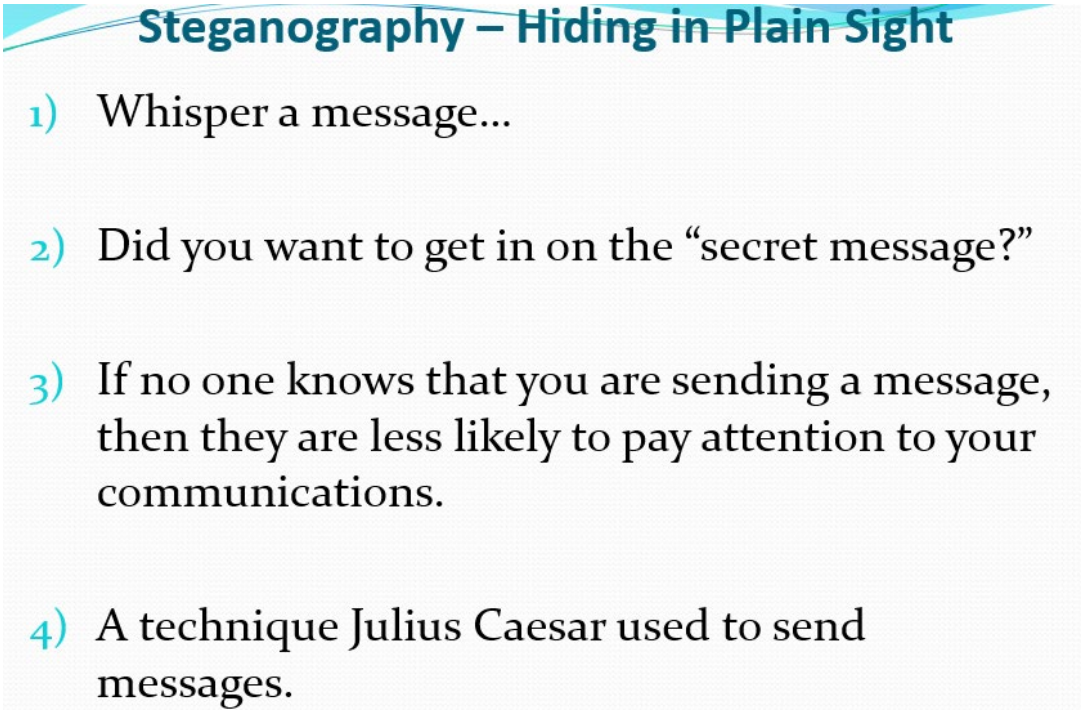
Calculate the check digit for the following UPC-12 code?

0 89126 74350 **?**

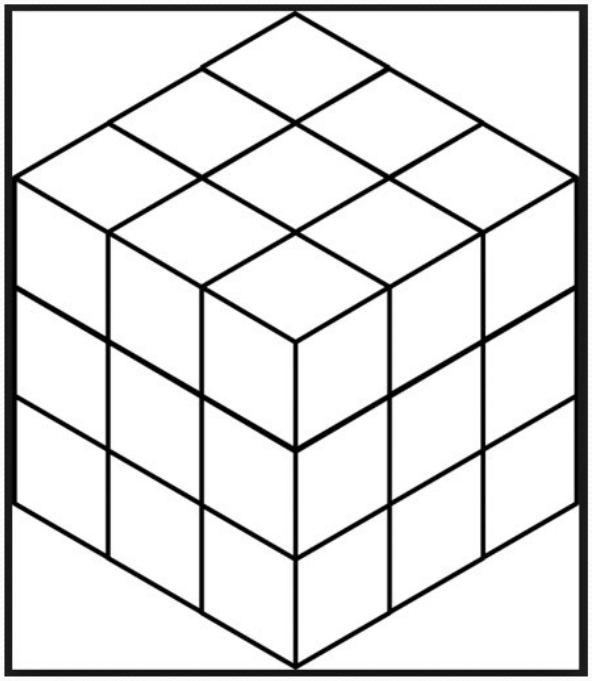
$$3 * (0 + 9 + 2 + 7 + 3 + 0) = 3 * (21) = 63$$
$$1 * (8 + 1 + 6 + 4 + 5) = 24$$

→ $(63 + 24) \% 10 = 87 \% 10 = 7$ → $10 - 7 = \mathbf{3}$

Besides overtly encrypting a secret message, another approach is "hiding in plain sight," or ***Steganography***. One of the first examples of this was from Julius Caesar. He would have a messenger shave their head, the message was tattooed on their head, and when their hair grew back, they were sent to deliver it. Not super timely, but pretty effective.

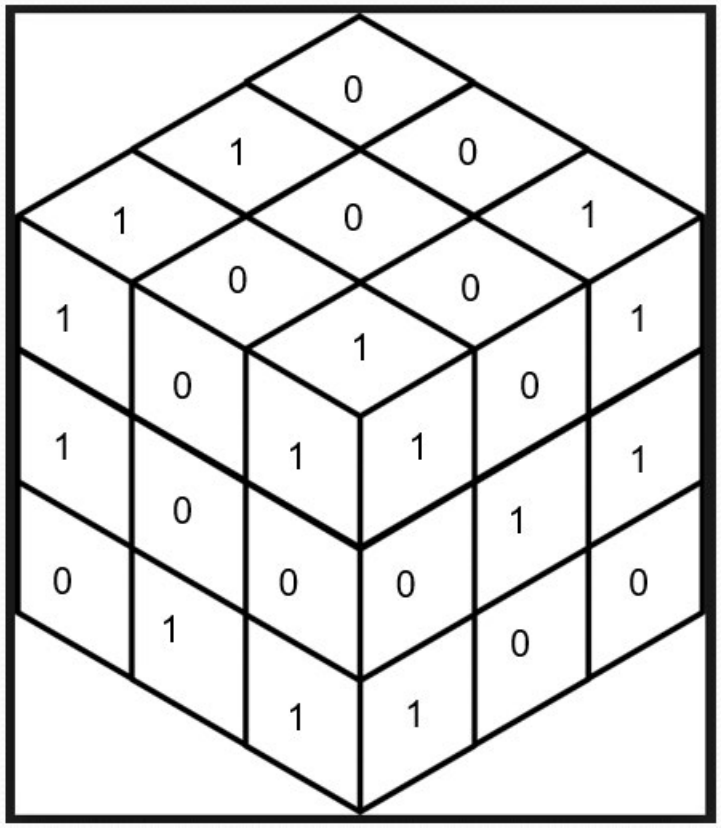


Steganography – Hiding in Plain Sight



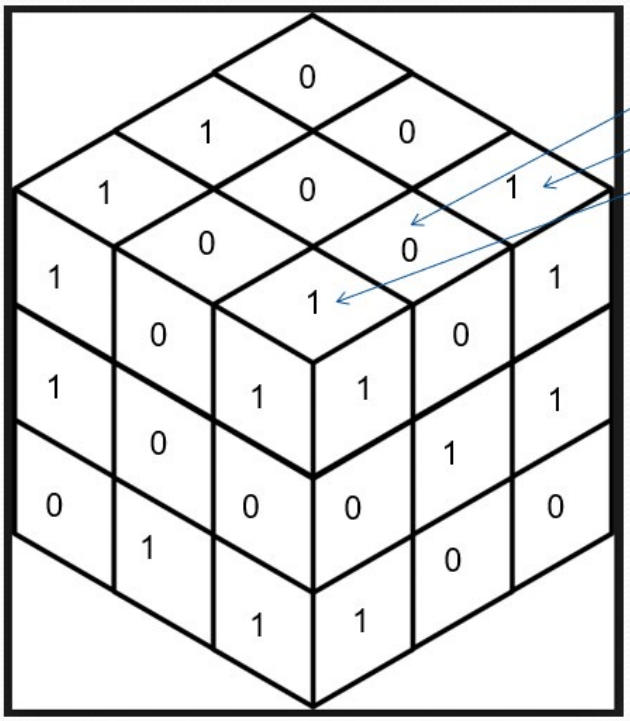
Steganography – Hiding in Plain Sight

Consider this representation of an image



Steganography – Hiding in Plain Sight

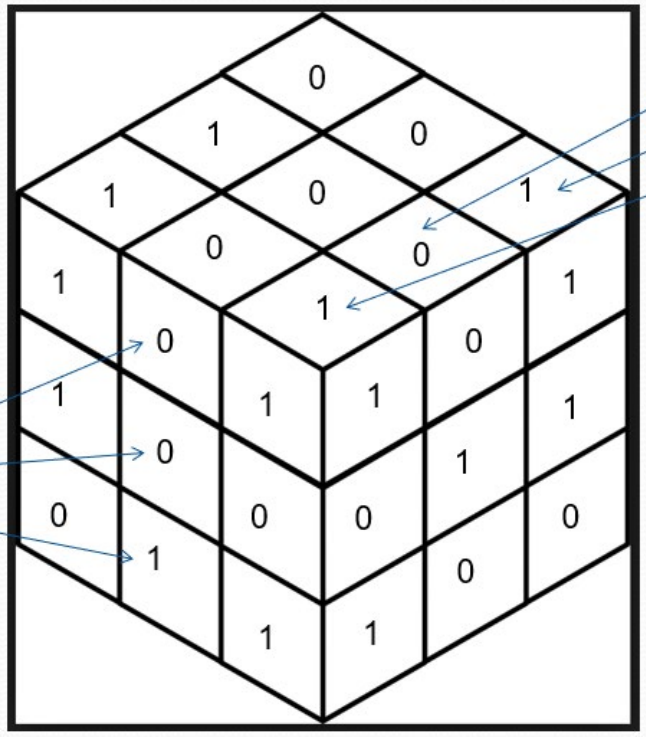
Consider this representation of an image



Each “column” is one pixel

Steganography – Hiding in Plain Sight

Consider this representation of an image

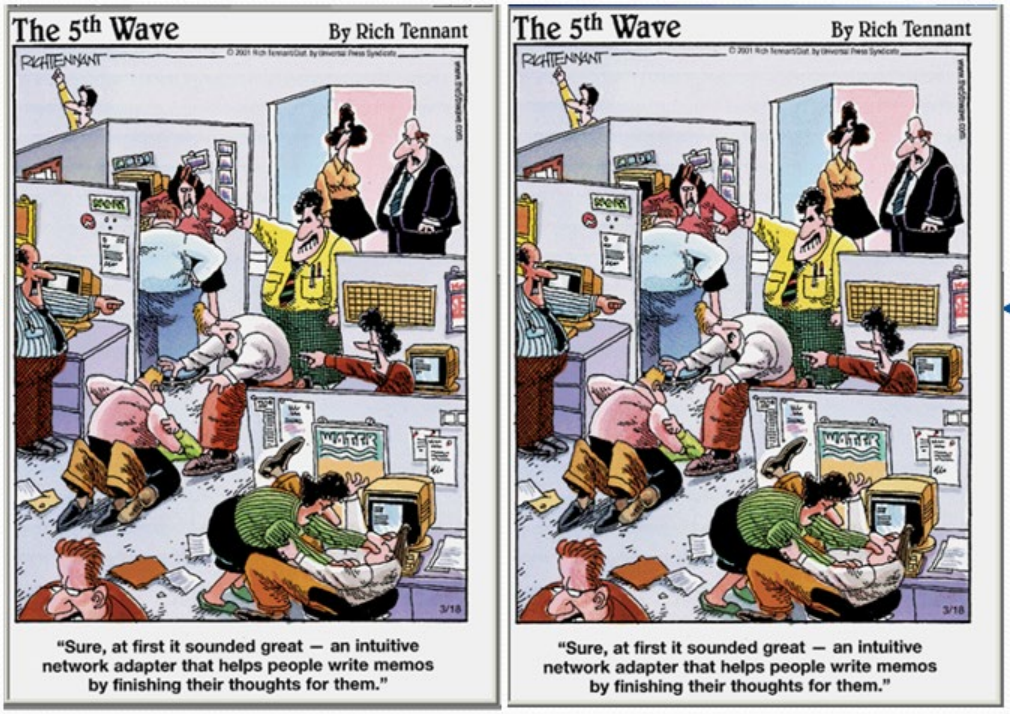


Each “column” is one pixel

If each color is 8 bits, then there would be 8 “layers”

If you remove the least significant "layer" of each pixel and replace it with a message, the image doesn't change much

Original image →



← Image with embedded text

If you remove the least significant "layer" of each pixel, and replace it with a message, the image doesn't change much

Original image →

Image of a tree with a steganographically hidden image. The hidden image is revealed by removing all but the two least significant bits of each color component and a subsequent normalization. The hidden image is shown below.

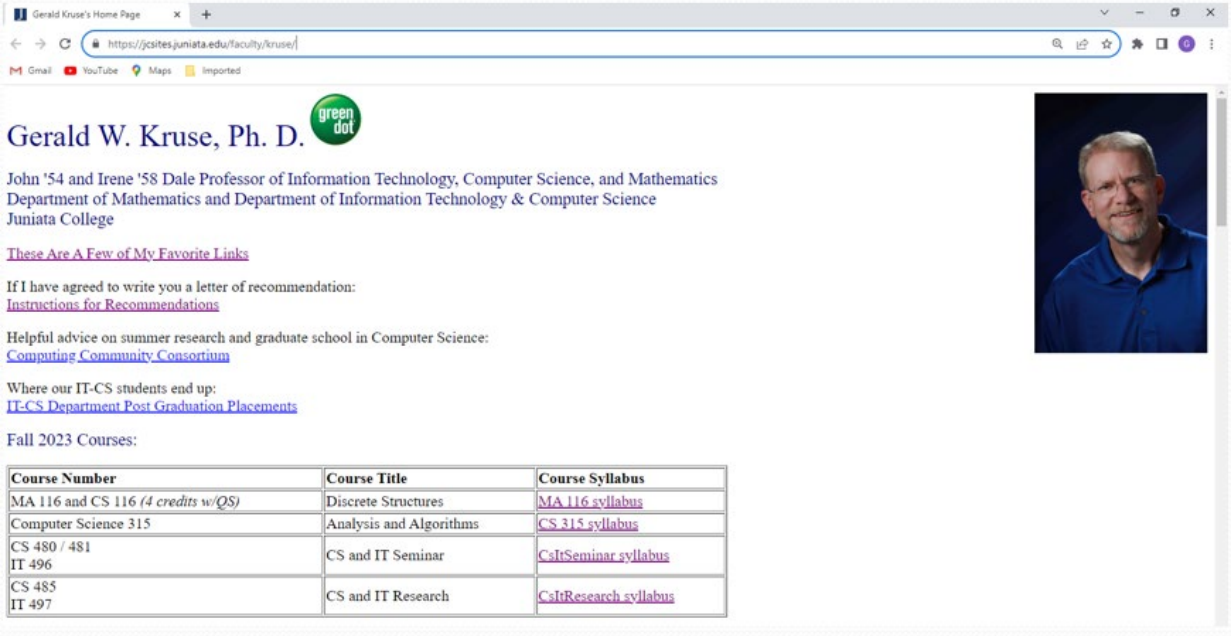
Image of a cat extracted from the tree image above.

← Embedded image

https://en.wikipedia.org/wiki/Steganography

faculty.juniata.edu/kruse

http://jcsites.juniata.edu/faculty/kruse/



Source code for jcsites.juniata.edu/faculty/kruse