CRYPTO M1 (part-2)

techworldthink • March 19, 2022

Classical Encryption techniques

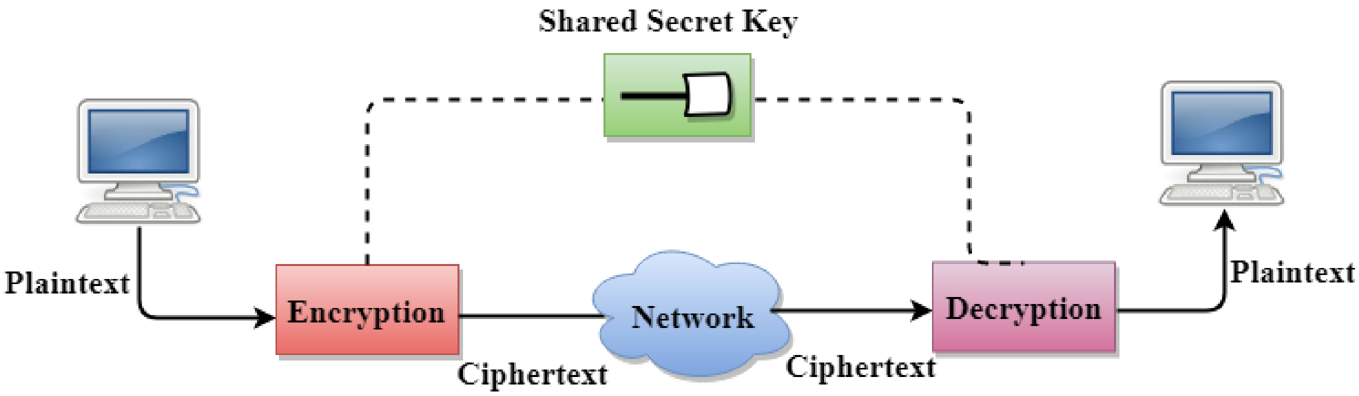
Symmetric cipher model

Substitution techniques

Transposition techniques

Symmetric cipher model

Symmetric encryption or shared key encryption is a method of encryption where both the parties involved share a standard key. That common key must be kept secret by both the parties. For example, “A” will encrypt a message with a shared key “K, ” then “B” can decrypt the encrypted message only with “K.”



Computer Network Privacy

- In Secret Key Encryption/Decryption technique, the same key is used by both the parties, i.e., the sender and receiver.
- The sender uses the secret key and encryption algorithm to encrypt the data; the receiver uses this key and decryption algorithm to decrypt the data.

- In Secret Key Encryption/Decryption technique, the algorithm used for encryption is the inverse of the algorithm used for decryption. It means that if the encryption algorithm uses a combination of addition and multiplication, then the decryption algorithm uses a combination of subtraction and division.
- The secret key encryption algorithm is also known as symmetric encryption algorithm because the same secret key is used in bidirectional communication.
- In secret key encryption/decryption algorithm, the secret code is used by the computer to encrypt the information before it is sent over the network to another computer.
- The secret key requires that we should know which computers are talking to each other so that we can install the key on each computer.

Substitution techniques

> *Caesar Cipher*

The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down.

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

(Encryption Phase with shift n)

$$E(x)=(x+n)mod\backslash\ 26$$

(Decryption Phase with shift n)

$$D(x)=(x-n)mod\backslash\ 26$$

Examples :

```
Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Shift: 23
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW
```

> *Playfair cipher*

The **Playfair cipher** was the first practical digraph substitution cipher. The scheme was invented in **1854** by **Charles Wheatstone** but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.

It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

1. **Generate the key Square(5×5):** The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

2. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

For example:

```
PlainText: "instruments"
After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'
```

1. Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.

Plain Text: *“hello” ,* ***After Split:*** *‘he’ ‘lx’ ‘lo’ , Here* ***‘x’*** *is the bogus letter.*

2. If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter

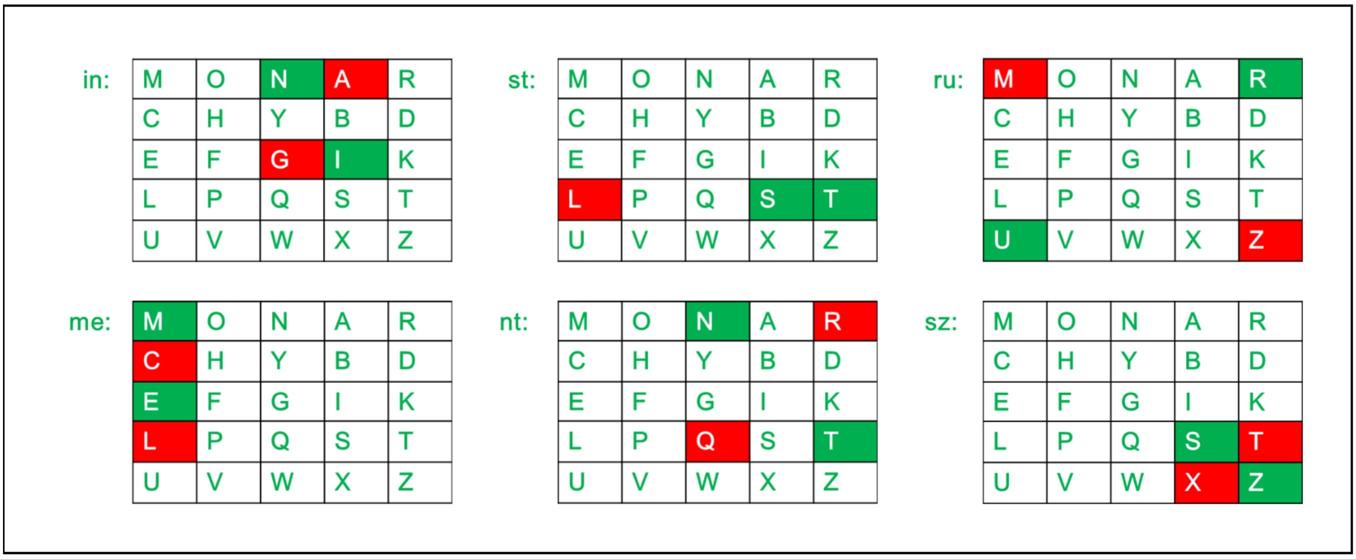
Plain Text: *“helloe” ,* ***AfterSplit:*** *‘he’ ‘lx’ ‘lo’ ‘ez’ , Here* ***‘z’*** *is the bogus letter.*

Rules for Encryption:

- **If both the letters are in the same column**: Take the letter below each one (going back to the top if at the bottom).
- **If both the letters are in the same row**: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

For example:

```
Plain Text: "instrumentsz"
Key : "MONARCHY"
Encrypted Text: gatlmzclrqtx
Encryption:
  i -> g
  n -> a
  s -> t
  t -> l
  r -> m
  u -> z
  m -> c
  e -> l
  n -> r
  t -> q
  s -> t
  z -> x
```



in:

M	O	🟩 N	🟥 A	R
C	H	Y	B	D
E	F	🟥 G	🟩 I	K
L	P	Q	S	T
U	V	W	X	Z

st:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
🟥 L	P	Q	🟩 S	🟩 T
U	V	W	X	Z

ru:

🟥 M	O	N	A	🟩 R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
🟩 U	V	W	X	🟥 Z

me:

🟩 M	O	N	A	R
🟥 C	H	Y	B	D
🟩 E	F	G	I	K
🟥 L	P	Q	S	T
U	V	W	X	Z

nt:

M	O	🟩 N	A	🟥 R
C	H	Y	B	D
E	F	G	I	K
L	P	🟥 Q	S	🟩 T
U	V	W	X	Z

sz:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	🟩 S	🟥 T
U	V	W	🟥 X	🟩 Z

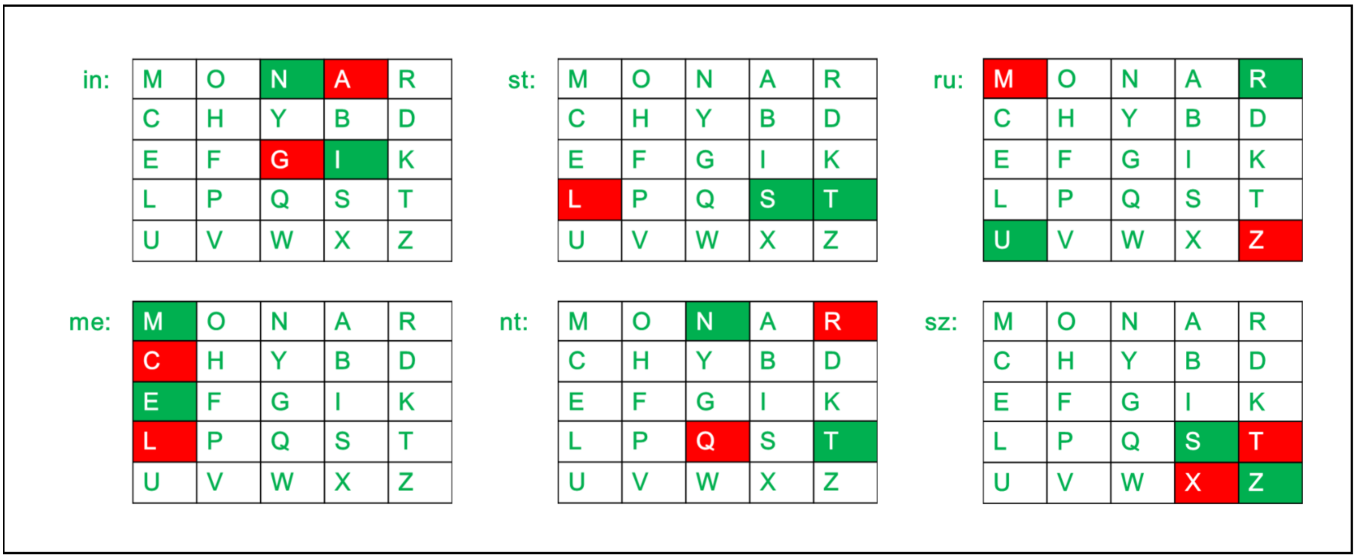
Example of encryption

Rules for Decryption:

- **If both the letters are in the same column**: Take the letter above each one (going back to the bottom if at the top).
- **If both the letters are in the same row**: Take the letter to the left of each one (going back to the rightmost if at the leftmost position).
- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

For example:

```
Plain Text: "gatlmzclrqtx"
Decrypted Text: instrumentsz
Decryption:
(red)-> (green)
  ga -> in
  tl -> st
  mz -> ru
  cl -> me
  rq -> nt
  tx -> sz
```



in:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

st:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

ru:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

me:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

nt:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

sz:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Example of Decryption

Hill cipher

Hill cipher is a polygraphic substitution cipher based on linear algebra.Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Examples:

```
Input  : Plaintext: ACT
         Key: GYBNQKURP
Output : Ciphertext: POH
.................................
```

```
Input  : Plaintext: GFG
         Key: HILLMAGIC
Output : Ciphertext: SWK
```

Encryption

We have to encrypt the message 'ACT' (n=3).The key is 'GYBNQKURP' which can be written as the nxn matrix:

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

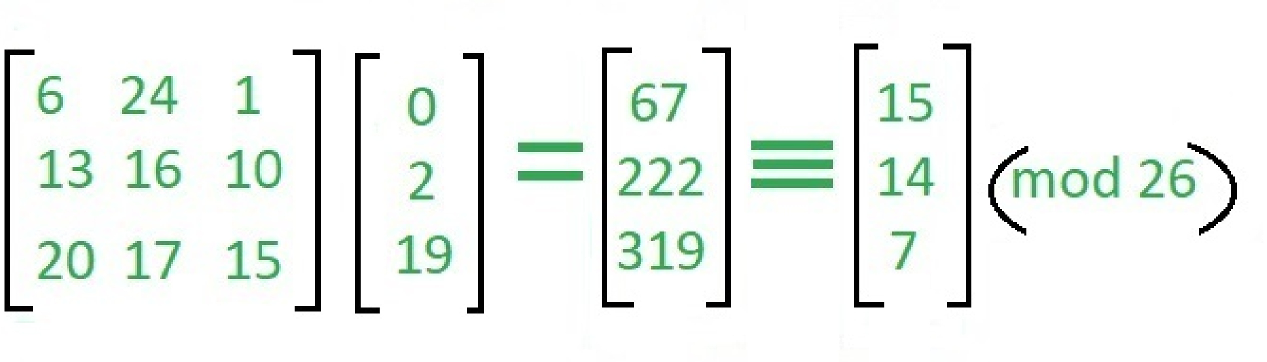
Cipherkey

The message 'ACT' is written as vector:

$$\begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}$$

message vector

The enciphered vector is given as:



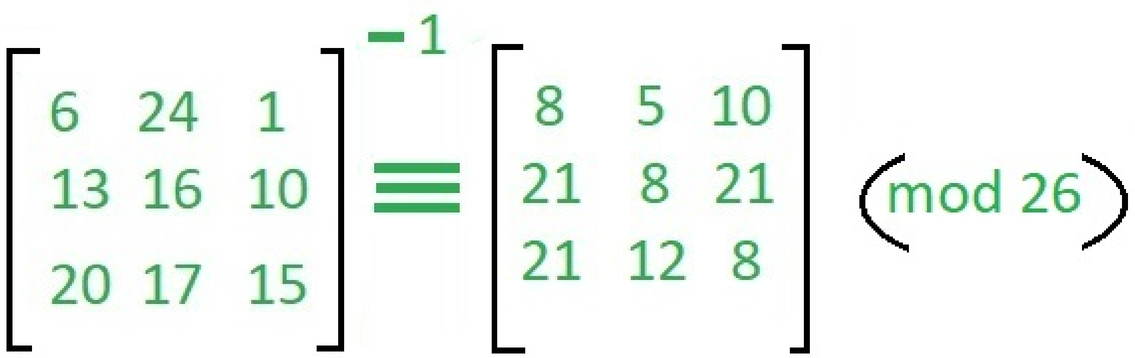
$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix} \equiv \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \pmod{26}$$

enciphered vector

which corresponds to ciphertext of ‘POH’

Decryption

To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters).The inverse of the matrix used in the previous example is:



$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}^{-1} \equiv \begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \pmod{26}$$

inverse matrix

For the previous Ciphertext ‘POH’:

$$\begin{bmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{bmatrix} \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \equiv \begin{bmatrix} 260 \\ 574 \\ 539 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix} \pmod{26}$$

Decrypt

which gives us back ‘ACT’.

Assume that all the alphabets are in upper case.

Transposition techniques

> *1. Rail-Fence Technique*

Rail-Fence is the simple Transposition technique that involves writing plain text as a sequence of diagonals and then reading it row by row to produce the ciphertext.

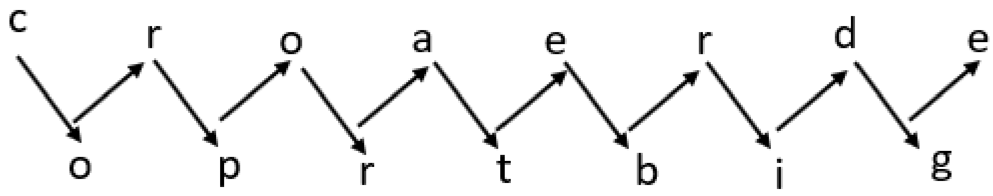
Algorithm

Step 1: Write down all the characters of plain text message in a sequence of diagnosis.

Step 2: Read the plain text written in step 1 as a sequence of rows.

Example: Suppose plain text corporate bridge, and we want to create the ciphertext of the given.

First, we arrange the plain text in a sequence of diagnosis, as shown below.



example

Now read the plain text by row-wise, i.e. croaerdeoprtbig.

So, here the plain text is a corporate bridge, and ciphertext is croaerdeoprtbig.

The Rail-Fence technique is quite easy to break.

2. Simple columnar transposition techniques

The simple columnar transposition technique can be categorized into two parts – Basic technique and multiple rounds.

Simples columnar transposition technique – basic technique. The simple columnar transposition technique simply arranges the plain text in a sequence of rows of a rectangle and reads it in a columnar manner.

Step 1: Write all the characters of plain text message row by row in a rectangle of predefined size.

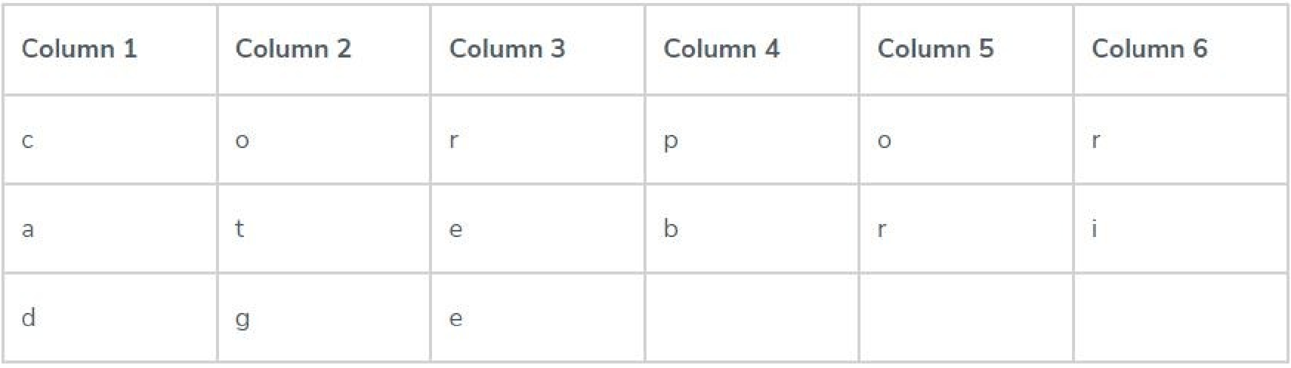
Step 2: Read the message in a columnar manner, i.e. column by column.

Note: For reading the message, it needs not to be in the order of columns. It can happen in any random sequence.

Step 3: The resultant message is ciphertext.

Example: Let’s assume that Plain text is a corporate bridge, and we need to calculate the cipher text using a simple columnar transposition technique.

Let’s take 6 columns and arrange the plain text in a row-wise manner.



Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
c	o	r	p	o	r
a	t	e	b	r	i
d	g	e			

Decide the column order for reading the message – let’s assume 1,3,5,2,4,6 is an order.

Now read the message in a columnar manner using the decided order. – cadreeorotgpbri

cadreeorotgpbri is a ciphertext.

Steganography

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data. The word *steganography* is derived from the Greek words *steganos* (meaning *hidden* or c*overed*) and the Greek root *graph* (meaning *to write*).

Steganography can be used to conceal almost any type of digital content, including text, image, video or audio content; the data to be hidden can be hidden inside almost any other type of digital content. The content to be concealed through steganography -- called *hidden text* -- is often encrypted before being incorporated into the innocuous-seeming *cover text* file or data stream. If not encrypted, the hidden text is commonly processed in some way in order to increase the difficulty of detecting the secret content.

Steganography is practiced by those wishing to convey a secret message or code. While there are many legitimate uses for steganography, malware developers have also been found to use steganography to obscure the transmission of malicious code.

How is steganography used today?

In modern digital steganography, data is first encrypted or obfuscated in some other way and then inserted, using a special algorithm, into data that is part of a particular file format such as a JPEG image, audio or video file. The secret message can be embedded into ordinary data files in many different ways. One technique is to hide data in bits that represent the same color pixels repeated in a row in an image file. By applying the encrypted data to this redundant data in some inconspicuous way, the result will be an image file that appears identical to the original image but that has "noise" patterns of regular, unencrypted data.

The practice of adding a watermark -- a trademark or other identifying data hidden in multimedia or other content files -- is one common use of steganography. Watermarking is a technique often used by online publishers to identify the source of media files that have been found being shared without permission.

While there are many different uses of steganography, including embedding sensitive information into file types, one of the most common techniques is to embed a text file into an image file. When this is done, anyone viewing the image file should not be able to see a difference between the original image file and the encrypted file; this is accomplished by storing the message with less significant bites in the data file. This process can be completed manually or with the use of a steganography tool.

What are the advantages of steganography over cryptography?

Steganography is distinct from cryptography, but using both together can help improve the security of the protected information and prevent detection of the secret communication. If steganographically-hidden data is also encrypted, the data may still be safe from detection -- though the channel will no longer be safe from detection. There are advantages to using steganography combined with encryption over encryption-only communication.

The primary advantage of using steganography to hide data over encryption is that it helps obscure the fact that there is sensitive data hidden in the file or other content carrying the hidden text. Whereas an encrypted file, message or network packet payload is clearly marked and identifiable as such, using steganographic techniques helps to obscure the presence of the secure channel.

Steganography software

Steganography software is used to perform a variety of functions in order to hide data, including encoding the data in order to prepare it to be hidden inside another file, keeping track of which bits of the cover text file contain hidden data, encrypting the data to be hidden and extracting hidden data by its intended recipient.

There are proprietary as well as open source and other free-to-use programs available for doing steganography. OpenStego is an open source steganography program; other programs can be characterized by the types of data that can be hidden as well as what types of files that data can be hidden inside. Some online steganography software tools include Xiao Steganography, used to hide secret files in BMP images or WAV files; Image Steganography, a Javascript tool that hides images inside other image files; and Crypture, a command line tool that is used to perform steganography.