**Tìm hiểu về phương pháp giấu tin trong ảnh và xây dựng thực nghiệm bằng Python**

Xayavong nalongsit

K13\_CNTT

1. **Cơ sở lý thuyết**
   1. **Định nghĩa và mục tiêu của phương pháp giấu tin trong ảnh**

Giấu tin trong ảnh là một kỹ thuật mã hóa thông tin vào trong một ảnh gốc. Thông tin được giấu được gọi là thông tin ẩn (embedded data). Mục tiêu của giấu tin trong ảnh là đảm bảo tính ẩn của thông tin ẩn, tức là người không biết về kỹ thuật giấu tin sẽ không thể phát hiện được sự tồn tại của thông tin ẩn. Ngoài ra, kỹ thuật giấu tin cần đảm bảo chất lượng của ảnh gốc không bị ảnh hưởng quá nhiều sau khi giấu tin.

* 1. **Các phương pháp phổ biến trong giấu tin ảnh**

Có nhiều phương pháp khác nhau để giấu tin trong ảnh. Một số phương pháp phổ biến bao gồm:

* LSB (Least Significant Bit) steganography: Phương pháp này chèn thông tin bí mật vào các bit ít quan trọng (LSB) của mỗi pixel trong ảnh.
* DCT (Discrete Cosine Transform) steganography: Phương pháp này chèn thông tin bí mật vào các biến đổi cosine rời rạc (DCT) của ảnh.
* SVD (Singular Value Decomposition) steganography: Phương pháp này chèn thông tin bí mật vào các giá trị phân tích thành phần riêng (SVD) của ảnh.
* Spread spectrum steganography: Phương pháp này chèn thông tin bí mật dưới dạng nhiễu trong ảnh.
  1. **Nguyên lý hoạt động của phương pháp LSB**

Nguyên lý hoạt động của phương pháp LSB (Least Significant Bit) steganography là ẩn thông tin trong các bit ít quan trọng nhất của dữ liệu gốc mà không làm thay đổi quá nhiều giá trị của dữ liệu đó.

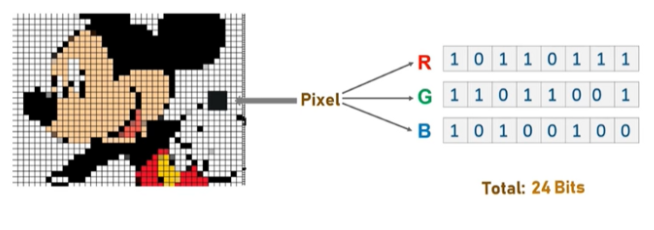
Dưới đây là một ví dụ sơ bộ để giải thích nguyên lý hoạt động của LSB steganography:

* Dữ liệu Gốc (Cover Data): Giả sử bạn có một hình ảnh, và mỗi pixel của hình ảnh đó được biểu diễn bởi một giá trị màu từ 0 đến 255 (trong hệ thống màu RGB).
* Thông Tin Cần Ẩn (Hidden Data): Bạn muốn ẩn một dãy bit thông tin trong hình ảnh mà không làm thay đổi quá nhiều.
* Thực Hiện Ẩn Thông Tin: Đối với mỗi pixel, bạn thay đổi bit ít quan trọng nhất của giá trị màu để chứa bit của thông tin bạn muốn giấu đi. Điều này có thể thực hiện bằng cách đặt bit ít quan trọng của giá trị màu bằng giá trị của bit thông tin cần giấu.

**Ví dụ:** Nếu bit ít quan trọng là 1 và bạn muốn giấu bit 0, bạn có thể thay đổi giá trị màu từ 10110101 thành 10110100.

* Quan sát và Phục Hồi: Khi bạn quan sát hình ảnh, sự thay đổi trong giá trị màu là rất nhỏ và thường không thể nhận biết bằng mắt thường. Người nhìn sẽ thấy hình ảnh không có sự thay đổi đáng kể, trong khi thông tin đã được giấu đi.
* Trích Xuất Thông Tin: Để trích xuất thông tin ẩn, bạn chỉ cần đọc bit ít quan trọng nhất của mỗi giá trị màu trong hình ảnh.

**Hình minh họa**

****

**A diagram of a number

Description automatically generated with medium confidence**

1. **Công cụ và Ngôn ngữ lập trình**
   1. **Python**

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

* **Đặc điểm đặc biệt của Python**

Python có một số điểm đặc biệt khiến nó trở nên nổi bật so với các ngôn ngữ lập trình khác, bao gồm:

* Dễ sử dụng và dễ học: Python sử dụng cú pháp đơn giản, dễ hiểu, giống như tiếng Anh
* Linh hoạt: Python là một ngôn ngữ đa mẫu hình, hỗ trợ cả lập trình hướng đối tượng và lập trình cấu trúc.
* Năng suất cao: Python có thể giúp các lập trình viên viết mã nhanh chóng và hiệu quả.
* **Ứng dụng của Python**

Python được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm:

* Khoa học dữ liệu và học máy: Python là một ngôn ngữ phổ biến cho khoa học dữ liệu và học máy.
* Trí tuệ nhân tạo: Python cũng là một ngôn ngữ phổ biến cho trí tuệ nhân tạo
* Phát triển web: Python là một lựa chọn phổ biến cho phát triển web.
* Phát triển ứng dụng di động: Python cũng có thể được sử dụng để phát triển ứng dụng di động.

1. **Thực nghiệm** 
   1. **Hiểu và triển khai thuật toán LSB**

**B1 nhập thư viện**

import numpy as np

import cv2

* Mã nhập thư viện NumPy dưới dạng npvà thư viện OpenCV dưới dạng cv2. Các thư viện này lần lượt được sử dụng cho các phép toán số và thị giác máy tính.

**B2 text\_to\_binary**

def text\_to\_binary(*msg*):

    if isinstance(*msg*, str):

        return ''.join([format(ord(i), "08b") for i in *msg*])

    elif isinstance(*msg*, (bytes, np.ndarray)):

        return [format(i, "08b") for i in *msg*]

    elif isinstance(*msg*, int) or isinstance(*msg*, np.uint8):

        return format(*msg*, "08b")

    else:

        raise TypeError("Input type is not supported in this function")

* Hàm này chuyển đổi văn bản hoặc dữ liệu nhị phân thành chuỗi nhị phân. Nó kiểm tra loại đầu vào ( **msg**) và chuyển đổi nó cho phù hợp. Đối với một chuỗi, nó chuyển đổi từng ký tự thành mã ASCII và sau đó thành biểu diễn nhị phân 8 bit.

**B3 encode\_image\_data**

def encode\_image\_data(*img*, *data*):

    if len(*data*) == 0:

        raise ValueError('Data entered to be encoded is empty')

    name\_of\_file = input("\nEnter the name of the New Image (Stego Image) after Encoding (with extension): ")

    no\_of\_bytes = (*img*.shape[0] \* *img*.shape[1] \* 3) // 8

    print("\t\nMaximum bytes to encode in Image:", no\_of\_bytes)

    if len(*data*) > no\_of\_bytes:

        raise ValueError("Insufficient bytes Error, Need Bigger Image or give Less Data !!")

*data* += '\*^\*^\*'

    binary\_data = text\_to\_binary(*data*)

    print("\n")

    print(binary\_data)

    length\_data = len(binary\_data)

    print("\nThe Length of Binary data", length\_data)

    index\_data = 0

    for i in *img*:

        for pixel in i:

            r, g, b = text\_to\_binary(pixel)

            if index\_data < length\_data:

                pixel[0] = int(r[:-1] + binary\_data[index\_data], 2)

                index\_data += 1

            if index\_data < length\_data:

                pixel[1] = int(g[:-1] + binary\_data[index\_data], 2)

                index\_data += 1

            if index\_data < length\_data:

                pixel[2] = int(b[:-1] + binary\_data[index\_data], 2)

                index\_data += 1

            if index\_data >= length\_data:

                break

    cv2.imwrite(name\_of\_file, *img*)

    print("\nEncoded the data successfully in the Image, and the image is successfully saved with name ", name\_of\_file)

Hàm này mã hóa dữ liệu nhị phân thành hình ảnh.

1. Nó kiểm tra xem độ dài của dữ liệu có trống hay không và nhắc người dùng tên của hình ảnh stego mới.
2. Nó tính toán số byte tối đa có thể được mã hóa trong hình ảnh dựa trên hình dạng của nó.
3. Nếu độ dài của dữ liệu vượt quá số byte tối đa cho phép thì sẽ xảy ra lỗi.
4. Dữ liệu được mã hóa được mở rộng với giá trị trọng điểm **'\*^\*^\*'**.
5. Biểu diễn nhị phân của dữ liệu mở rộng có được bằng cách sử dụng **text\_to\_binary** hàm.
6. Hàm lặp qua từng pixel trong ảnh, sửa đổi bit có ý nghĩa nhỏ nhất của mỗi kênh màu (R, G, B) bằng bit tương ứng từ dữ liệu nhị phân.
7. Hình ảnh được mã hóa sẽ được lưu.

**B4 decode\_image\_data**

def decode\_image\_data(*img*):

    data\_binary = ""

    for i in *img*:

        for pixel in i:

            r, g, b = text\_to\_binary(pixel)

            data\_binary += r[-1]

            data\_binary += g[-1]

            data\_binary += b[-1]

            total\_bytes = [data\_binary[i: i + 8] for i in range(0, len(data\_binary), 8)]

            decoded\_data = ""

            for byte in total\_bytes:

                decoded\_data += chr(int(byte, 2))

                if decoded\_data[-5:] == "\*^\*^\*":

                    print("\n\nThe Encoded data which was hidden in the Image was: ", decoded\_data[:-5])

                    return

Hàm này giải mã dữ liệu nhị phân từ hình ảnh.

1. Nó lặp qua từng pixel trong ảnh, trích xuất bit có ý nghĩa nhỏ nhất từ ​​​​mỗi kênh màu (R, G, B).
2. Các bit được trích xuất được nối với nhau để tạo thành biểu diễn nhị phân của dữ liệu ẩn.
3. Dữ liệu nhị phân được chia thành các khối 8 bit để biểu thị byte.
4. Nó chuyển đổi từng byte thành mã ASCII tương đương và kiểm tra giá trị canh gác '\*^\*^\*'để xác định phần cuối của tin nhắn ẩn.

**B5 main**

def main():

    while True:

        print("\n\t\tIMAGE STEGANOGRAPHY OPERATIONS\n")

        print("1. Encode the Text message")

        print("2. Decode the Text message")

        print("3. Exit")

        choice = int(input("Enter the Choice: "))

        if choice == 1:

            image = cv2.imread("./asset/img.jpg")

            data = input("\nEnter the data to be Encoded in Image: ")

            encode\_image\_data(image, data)

        elif choice == 2:

            image\_path = input("Enter the Image you need to Decode to get the Secret message: ")

            image = cv2.imread(image\_path)

            decode\_image\_data(image)

        elif choice == 3:

            break

        else:

            print("Incorrect Choice")

        print("\n")

Chức năng chính cung cấp giao diện dựa trên bảng điều khiển đơn giản cho các hoạt động steganography:

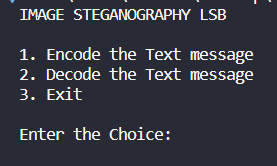
1. Người dùng được cung cấp các tùy chọn để mã hóa, giải mã hoặc thoát.
2. Nếu người dùng chọn mã hóa, nó sẽ tải một hình ảnh, lấy dữ liệu đầu vào của người dùng để mã hóa và gọi hàm encode\_image\_data.
3. Nếu người dùng chọn giải mã, nó sẽ lấy đường dẫn hình ảnh làm đầu vào, tải hình ảnh và gọi hàm decode\_image\_data.
4. Nếu người dùng chọn thoát thì chương trình sẽ kết thúc.

**B6 Chấp hành**

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Điều này đảm bảo rằng mainhàm chỉ được gọi khi tập lệnh được thực thi trực tiếp

* 1. **kết quả thức nghiệm**

**• Encoding the message in image file**

**A screenshot of a computer

Description automatically generated**

A camel in the desert

Description automatically generated**A camel in the desert

Description automatically generated• Preview of Cover image and Stego image:**

**• Decoding the message from Image file:**

**A screen shot of a computer

Description automatically generated**

1. **Kết luận**
   1. **Tóm tắt kết quả nghiên cứu**

Phương pháp Steganography LSB (Least Significant Bit) là một trong những phương pháp đơn giản nhất để giấu thông tin vào trong một hình ảnh. Dưới đây là một tóm tắt về kết quả nghiên cứu và đặc điểm chính của phương pháp này:

1. Nguyên tắc hoạt động:
   * Trong ảnh màu RGB, mỗi pixel thường được biểu diễn bởi 3 kênh màu: đỏ (R), xanh lục (G), và xanh lam (B).
   * Phương pháp LSB tận dụng việc chỉ có một số ít bit cuối cùng của mỗi kênh màu ảnh thường xuyên thay đổi khi thay đổi giá trị của pixel, nhất là khi độ chênh lệch giữa giá trị pixel mới và pixel ban đầu là nhỏ.
2. Ưu điểm:
   * Dễ triển khai: Phương pháp này rất dễ triển khai và hiểu, không đòi hỏi nhiều tài nguyên tính toán.
   * Ẩn dụ nhẹ: Thông tin được giấu nhẹ nhàng và không gây thay đổi rõ ràng trong hình ảnh.
3. Nhược điểm:
   * Dễ phát hiện: Phương pháp LSB dễ bị phát hiện vì sự thay đổi ít lớn ở bit thấp của mỗi kênh màu có thể dễ dàng được phát hiện bằng các phương pháp phân tích thống kê.
   * Giả mạo: Các kẽm giả mạo (mọi thay đổi nhỏ trong hình ảnh) có thể làm mất mát thông tin quan trọng hoặc gây nhiễu loạn.
4. Ứng dụng:
   * Bảo mật thông tin: LSB thường được sử dụng để giấu thông tin như mật khẩu hoặc các dữ liệu nhạy cảm trong hình ảnh mà không thu hút sự chú ý.
   * Chứng thực: Có thể được sử dụng để chứng minh nguồn gốc của một hình ảnh hoặc xác nhận tính toàn vẹn của dữ liệu.
   1. **phát triển tương lai**

* Kết hợp các phương pháp mới: Nghiên cứu liên quan đến việc kết hợp LSB với các phương pháp khác để tạo ra các kỹ thuật steganography mạnh mẽ hơn , khó phát hiện hơn và ngiện cứu thêm về giấu tin trong text,ẩm thánh,vdo

1. **Tài liệu tham khảo**
2. [**https://medium.com/@renantkn/lsb-steganography-hiding-a-message-in-the-pixels-of-an-image-4722a8567046**](https://medium.com/@renantkn/lsb-steganography-hiding-a-message-in-the-pixels-of-an-image-4722a8567046)
3. [**https://vi.wikipedia.org/wiki/Python\_(ng%C3%B4n\_ng%E1%BB%AF\_l%E1%BA%ADp\_tr%C3%ACnh)**](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh))
4. [**https://viblo.asia/p/ki-thuat-an-ma-dua-tren-cac-vi-tri-cua-lsb-demo-vyDZOkoOZwj**](https://viblo.asia/p/ki-thuat-an-ma-dua-tren-cac-vi-tri-cua-lsb-demo-vyDZOkoOZwj)
5. [**https://luanvan.co/luan-van/giau-tin-trong-anh-bang-ma-hoa-lsb-34362/**](https://luanvan.co/luan-van/giau-tin-trong-anh-bang-ma-hoa-lsb-34362/)