# **Ẩn lượng lớn dữ liệu bằng cách sử dụng các tiếp cận mới của giấu tin trong Video**

## **1.Đánh số dựa trên dãy hỗn loạn**

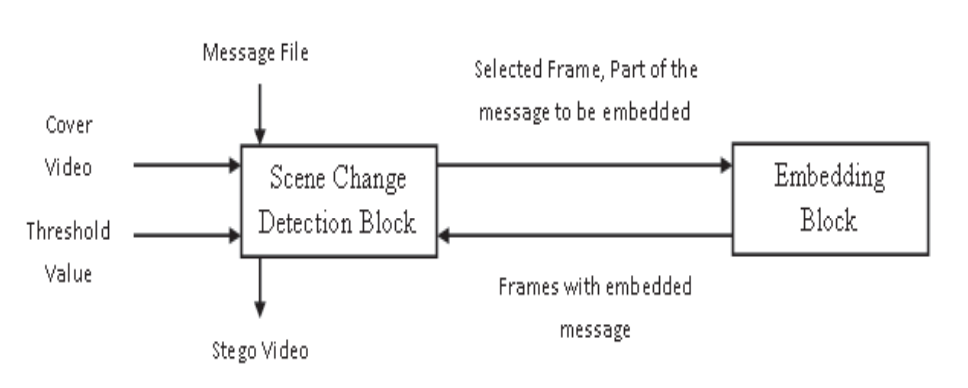
Hàm dãy hỗn loạn là hàm sinh ra các dãy ngẫu nhiên để tạo dãy giả tự nhiên bằng cách sử dụng phương trình sau:

 (1)

Phương trình (1) tạo ra một dãy số thực ngẫu nhiên. Nhưng để quyết định vị trí pixel cần yêu cầu số nguyên. Dãy giá trị thực có thể được chuyển đổi thành dãy số nguyên bằng cách sử dụng ý tưởng đánh số dựa trên dãy hỗn loạn. Trong cách tiếp cận này, một dãy logistic 1D được tạo ra sử dụng (1) và sau đó chỉ số của các phần tử trong dãy ban đầu được sử dụng như một dãy riêng biệt. Dãy mới này hiện bao gồm các giá trị số nguyên.

## 2.Mô hình đề xuất

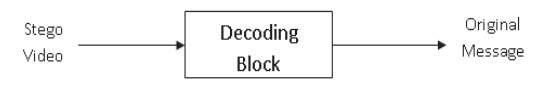
Mô hình được đề xuất bao gồm hai giai đoạn - giai đoạn nhúng và giai đoạn giải mã. Giai đoạn nhúng bao gồm hai khối là khối phát hiện thay đổi cảnh và khối nhúng. Khối phát hiện thay đổi cảnh kiểm tra từng khung hình để tìm sự thay đổi cảnh đột ngột bằng cách sử dụng kỹ thuật chênh lệch biểu đồ. Nó cũng chia thông điệp thành số khối tùy thuộc vào khả năng nhúng của một khung hình trong chuỗi video.



Hình 1. Sơ đồ khối của giai đoạn nhúng

Khi một khung được kiểm tra và xác nhận về sự thay đổi cảnh đột ngột, nó sẽ được gửi cùng với khối thông báo được nhúng vào khối nhúng. Khối nhúng nhúng khối thông báo vào khung bằng cách sử dụng cách tiếp cận 3–3–2 của kỹ thuật thay thế Bit ít có ý nghĩa nhất (LSB). Trong cách tiếp cận này, ký tự thông báo được chuyển đổi thành dòng bit 8 bit của nó. Một pixel sau đó được lấy từ cùng một vị trí trong mặt phẳng đỏ, mặt phẳng xanh lá cây và mặt phẳng xanh lam của khung hình. Sau đó, 3 LSB của pixel từ mặt phẳng màu đỏ được thay thế bằng 3 bit đầu tiên của dòng bit ký tự thông báo, 3 LSB của pixel từ mặt phẳng màu xanh lá cây được thay thế bằng 3 bit tiếp theo của dòng bit ký tự và 2 LSB của pixel từ mặt phẳng màu xanh lam được thay thế bằng 2 bit cuối cùng của dòng bit ký tự thông báo.

Để biểu thị sự kết thúc của khối tin nhắn, một ký tự NULL được thêm vào khối tin nhắn dưới dạng dấu phân cách. Ngoài ra, vị trí pixel được ngẫu nhiên hóa bằng cách sử dụng chuỗi số nguyên được tạo bằng cách sử dụng đánh số dựa trên dãy hỗn loạn để tránh nhúng các bit thông báo vào pixel một cách tuần tự. Sau khi toàn bộ thông báo đã được nhúng, id của khung được sử dụng để nhúng sẽ được nhúng vào khung đầu tiên của chuỗi video.



Hình 2. Sơ đồ khối của đoạn giải mã

Giai đoạn giải mã bao gồm một khối giải mã. Khối giải mã đầu tiên trích xuất tất cả các id khung được nhúng trong khung đầu tiên của chuỗi video và sau đó sử dụng các id này để trích xuất các bit thông báo từ các khung. Các bit thông báo này được nối, chuyển đổi thành định dạng ký tự của chúng và nối với nhau để tạo thành thông báo gốc sau khi loại bỏ ký tự NULL trong khối thông báo khác nhau. Các thuật toán được thiết kế cho mô hình này như sau.

## **2.Thuật toán phát hiện thay đổi khung cảnh**

Thuật toán sau phát hiện các khung trong video (vật chứa) nơi diễn ra sự thay đổi cảnh và gọi hàm nhúng cho từng khung như vậy cho đến khi toàn bộ thông điệp được nhúng.

Thuật toán như sau:

1. Đưa video vật chứa ‘v’ có:

F: Tổng số khung hình

rows: Chiều cao khung hình

cols: Chiều rộng khung hình

1. Chia nhỏ thông điệp m thành các khối m’ [low\_limit to up\_limit] dựa trên lượng dữ liệu được nhúng vào 1 khung duy nhất
2. 



1. for i=1 to (F-1) do
   1. 



* 1. for  do

for  do

1. 
2. 
3. , 
4. 
   1. Tìm giá trị trung bình (mean) của b cho tất cả các khối trong khung.
   2. If ( mean > threshold AND up\_limit <= length of message ) then do
      * 1. Lưu  trong mảng 
        2. Gọi hàm nhúng ([*low\_limit* to *up\_limit*],8)
        3. Thay thế khung ban đầu bằng khung đã sửa đổi
5. Gọi hàm nhúng 

## **3.Thuật toán để nhúng thông điệp: Hàm nhúng (frame(khung), array(mảng), b(số lượng bit)**

Thuật toán sau được sử dụng để nhúng thông báo. Hàm nhúng được gọi mỗi khi phát hiện một khung hình có thay đổi cảnh diễn ra. Hàm nhúng còn được gọi là lần cuối để nhúng mảng chứa id của các khung mà sự thay đổi cảnh được phát hiện và mảng này được nhúng vào khung đầu tiên vì khung này không được sử dụng trong quá trình nhúng thông báo thực tế.

Thuật toán như sau:

1. Tạo chuỗi số nguyên ngẫu nhiên sử dụng “Đánh số dựa trên dãy hỗn loạn”(1) sử dụng  và như là giá trị ban đầu và lưu dãy trong mảng ‘a’
2. arrayarray + NULL character
3. Trích xuất các vùng RED, GREEN, BLUE của khung
4. for  to length of array
   1. Chọn 1 pixel màu RED, GREEN, BLUE tại vị trí a(i)
   2. Chọn thông điệp tại vị trí thứ i và chuyển đổi thành b bits
   3. Thay thế các bit trong mặt phẳng LSB của các pixel trong mặt phẳng RED, GREEN, BLUE bằng các bit của các phần tử trong mảng.
5. Nhóm các mặt phẳng RED, GREEN, BLUE đã sửa đổi của khung và gửi nó trở lại khối phát hiện thay đổi cảnh.

## **4. Thuật toán giải mã video giấu tin**

Thuật toán sau đây là để giải mã video giấu tin để trích xuất tin nhắn gốc. Đầu tiên, các id khung được trích xuất từ khung đầu tiên và sau đó thông báo được trích xuất từ các khung được xác định bởi id khung.

Thuật toán như sau:

1. Video giấu tin ‘v’ có:

F: Tổng số khung hình

rows: Chiều cao khung hình

cols: Chiều rộng khung hình

1. Tạo một chuỗi số nguyên bằng cách sử dụng đánh số logistic map [] sử dụng các giá trị giống như được sử dụng trong quá trình nhúng và lưu dãy trong mảng ‘a’
2. 
3. Trích xuất các vùng RED, GREEN, BLUE của khung 
4. for  to leght of *temp* do
   1. 
   2. Lấy pixel tại vị trí *id* từ các vùng RED, GREEN, BLUE của khung 
   3. Trích xuất 4 LSBs mỗi điểm từ mặt phẳng pixcel RED, pixcel GREEN, pixcel BLUE của khung f.
   4. Kết hợp 12 bits và tìm giá trị nguyên x
   5.  cho đến khi gặp NULL
5. for i=1 to length of *frameid* do
   1. 
   2. 
   3. for  to length of *f1*
      1. *id a(i)*
      2. Lấy pixel tại vị trí *id* từ các vùng RED, GREEN, BLUE của khung 
      3. Trích xuất 3 LSBs từ mặt phẳng pixcel RED, 3 LSBs từ mặt phẳng pixcel GREEN, 2 LSBs từ mặt phẳng pixcel BLUE của khung *f1*.
      4. Kết hợp 8bits và tìm ký tự tương đương *ch.*
6. original\_message  original\_message + temp\_msg

## **5. Khả năng lưu chữ dữ liệu**

Tổng lượng dữ liệu mà một khung video có thể chứa thể hiện khả năng nhúng của khung.

Với:

m  số dòng trong frame (height of the frame).

n  số cột trong frame (width of the frame).

Từ đó kích thước mỗi frame là *m\*n*

Mỗi khung hình trong video màu được tạo bởi ba thành phần tức là RED,GREEN và BLUE.

Do đó, tổng số pixel trong một khung hình là *m\*n\*3*

Trong mô hình này, 8 bit của mỗi ký tự thông điệp được nhúng ở định dạng

3: 3: 2 tức là 3 bit đầu tiên trong 3 LSBs của pixel trong thành phần màu đỏ, 3 bit tiếp theo trong 3 LSBs của pixel trong thành phần màu xanh lá cây và cuối cùng 2 bit trong 2 LSBs của pixel trong thành phần màu xanh lam. Vì vậy, để nhúng 8 bit,

chúng tôi yêu cầu 24 bit tức là ba pixel 8 bit.

Vì vậy, khả năng nhúng của mô hình được đề xuất là *m\*n*

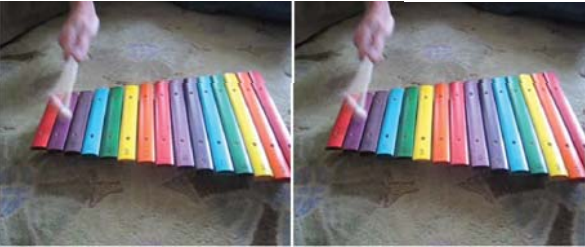
## **6. Kết quả thực nghiệm và phân tích hiệu suất**

Thuật toán được đề xuất đã được thử nghiệm với ba tệp video - cola.wmv, xylo.mpg và thoi.avi dưới dạng video vật chứa và 'ccs.txt' là tệp tin nhắn riêng biệt. Video đầu ra 'colaout.avi', 'xyloout.avi' và thoiout.avi 'được tạo là video giấu tin của các video vật chứa tương ứng của chúng.

Các hình bên dưới cho thấy các khung của video vật chứa và các khung của video giấu tin mà trong đó thông điệp đã được nhúng vào. Đầu ra cho thấy hoàn toàn không có sự khác biệt về mặt cảm quan giữa video vật chứa và video giấu tin.



Hình 3. Số khung 28 của Video vật chứa ‘cola.wmv’ và Video giấu tin ‘colaout.avi’



Hình 4. Số khung 11 của Video vật chứa ‘xylo.mpg’ và Video giấu tin ‘xyloout.avi’

Chất lượng cảm nhận của khung video bìa và khung video vật chứa đã được đánh giá bằng cách sử dụng hai số liệu đánh giá chất lượng - Sai số toàn phương trung bình (MSE) và Tỷ lệ tín hiệu trên tiếng ồn cao nhất (PSNR).

6.1. Sai số toàn phương trung bình (MSE)

Sai số toàn phương trung bình (MSE) được sử dụng để xác định khung video vật chứa và khung video giấu tin khác nhau như thế nào. Điều này được thực hiện bằng cách lấy tổng chênh lệch giữa các giá trị pixel tương ứng của cả hai khung và sau đó chia tổng cho kích thước của khung. Kể từ MSE xác định mức độ khác biệt giữa khung vật chứa và khung giấu tin, do đó, giá trị của MSE thấp hơn là kết quả.

MSE = (2)

Ở đâu *M* và *N* là các hàng và cột của khung video, *f (m,n)* là giá trị pixel ở vị trí *(m,n)* bên trong bao gồm khung video vật chứa và *F(m, n)* là giá trị pixel tại vị trí *(m,n)* trong khung video giấu tin tương ứng.

6.2. Tỉ số tín hiệu cực đại trên nhiễu( PSNR)

Tỉ số tín hiệu cực đại trên nhiễu (PSNR) được sử dụng để xác định khung video vật chứa và khung video giấu tin tương ứng giống nhau đến mức nào. Vì PSNR xác định mức độ giống nhau giữa khung vật chứa và khung giấu tin, do đó kết quả là giá trị PSNR càng cao càng tốt.

PSNR=

Ở đâu R là giá trị tối đa có thể có của độ chói. Đối với giá trị hình ảnh 8 bit của R sẽ là 255. PSNR được đo bằng decibels (dB).

Các giá trị MSE và PSNR cho khung video bìa và khung video stego được hiển thị trong bảng sau:

Bảng 1 Kết quả đánh giá chất lượng khung video vật chứa và khung video giấu tin.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Video vật chứa | Video giấu tin | Số khung | MSE | PSNR(dB) |
| cola.wmv | colaout.avi | 28 | 1.92 | 141.65 |
| 99 | 0.70 | 146.04 |
| xylo.mpg | xyloout.avi | 11 | 2.00 | 141.47 |
| 12 | 1.97 | 141.53 |
| shuttle.avi | shuttleout.avi | 81 | 2.03 | 141.41 |
| 84 | 0.08 | 155.29 |

6.3. Sự khác biệt trên biểu đồ tần suất