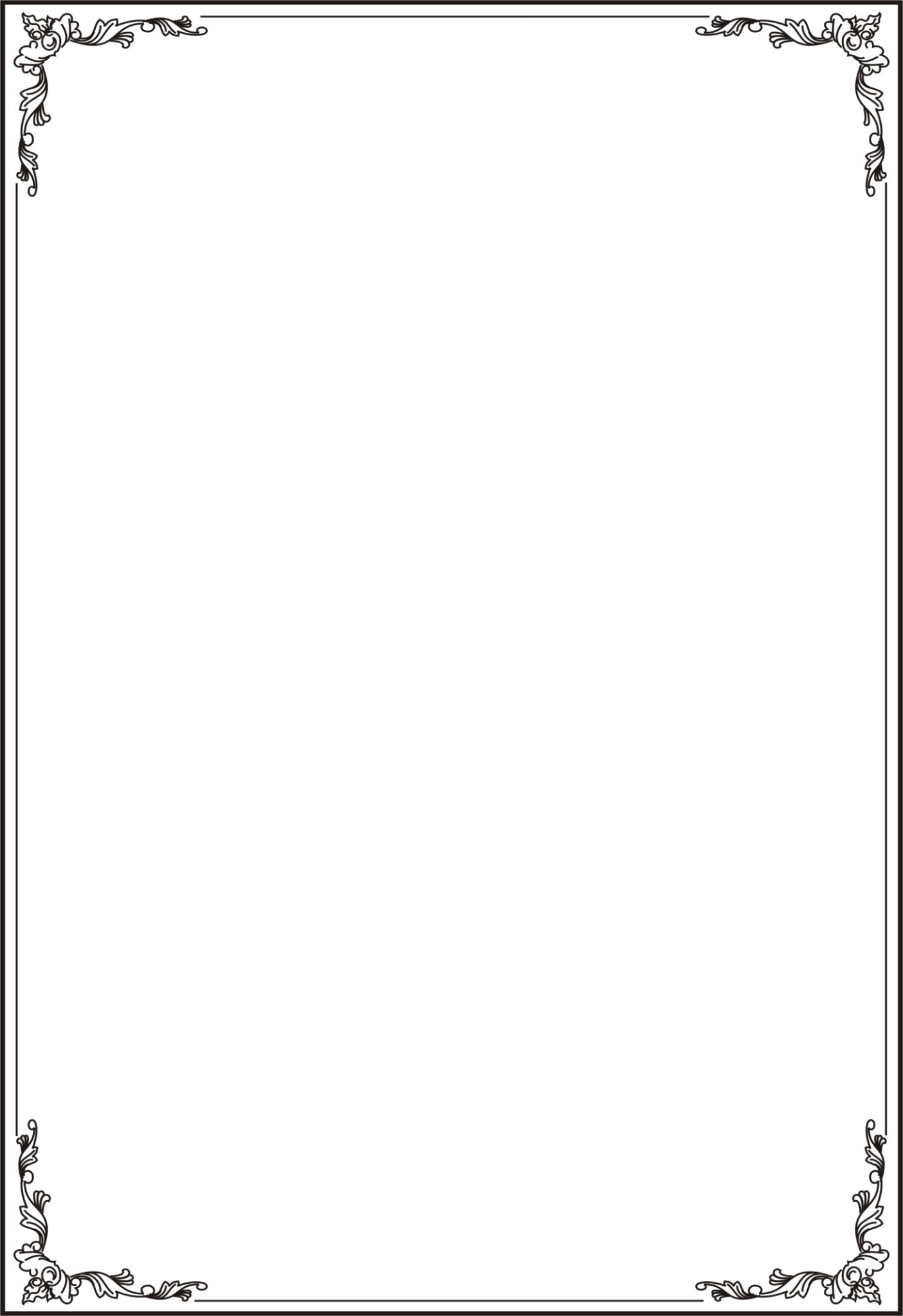
****

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**-----------------------------**



**BÁO CÁO TỐT NGHIỆP**

***Môn học: AN TOÀN PHẦN MỀM***

Chủ đề: PHÁT HIỆN GIẤU TIN TRONG VIDEO  
(VIDEO STEGANASLYSIS)

**Giảng viên : Đỗ Xuân Chợ**

Nhóm sinh viên : Tô Duy Nghĩa

Đỗ Nguyễn Tuấn

Hà Nội 2018

Mục Lục

[**A- LÝ THUYẾT PHÁT HIỆN GIẤU TIN TRONG VIDEO** 3](#_Toc528377815)

[***I- Tổng quan phát hiện giấu tin trong video.*** 3](#_Toc528377816)

[***II- Phát hiện dựa theo đối tượng mang tin.*** 5](#_Toc528377817)

[***1. Phát hiện giấu tin dựa trên Exploring the Temporal Correlation between Frames (mối tương quan về thời gian giữa các khung hình).*** 5](#_Toc528377818)

[***2. Phát hiện giấu tin trong video dựa trên Asymptotic Relative Efficiency(ARE).*** 7](#_Toc528377819)

[***3. Phát hiện giấu tin trong video dựa trên Mode Detection.*** 7](#_Toc528377820)

[***4. Phát hiện giấu tin trong video dựa trên Spatial and Temporal Prediction (dự đoán trên miền không gian và thời gian).*** 8](#_Toc528377821)

[***5. Các kỹ thuật phát hiện giấu tin khác*** 8](#_Toc528377822)

[***III- Phát hiện dựa theo thuật toán giấu tin*** 8](#_Toc528377823)

[***1. Tệp hình ảnh.*** 8](#_Toc528377824)

[***2. Tập tin âm thanh.*** 9](#_Toc528377825)

[**B. NGHIÊN CỨU: STEGANALYSIS - PHÁT HIỆN GIẤU TIN** 10](#_Toc528377826)

[***I. Sơ lược về Steganalysis*** 10](#_Toc528377827)

[***II. Đặt vấn đề*** 12](#_Toc528377828)

[***III. Xây dựng thuật toán*** 14](#_Toc528377829)

[***1. Thuật toán*** 14](#_Toc528377830)

[***2. Collusion attack (Tấn công tích hợp)*** 16](#_Toc528377831)

[***3. Pattern Classifier*** 26](#_Toc528377832)

[***IV. Kết luận*** 30](#_Toc528377833)

[**C. DEMO** 32](#_Toc528377834)

[***Tài Liệu Tham Khảo*** 33](#_Toc528377835)

# 

# **A- LÝ THUYẾT PHÁT HIỆN GIẤU TIN TRONG VIDEO**

## ***I- Tổng quan phát hiện giấu tin trong video.***

Steganalysis là kỹ thuật phát hiện thông tin ẩn giấu trong multimedia. Mục đích phát hiện ra thông tin ẩn và phá vỡ tính bí mật của vật mang tin ẩn. Ứng dụng trực tiếp các kỹ thuật phát hiện giấu tin trong hình ảnh vào video trên cơ sở từng khung hình không đem lại hiệu quả cao. Giống như phát hiện giấu tin trong âm thanh, có rất ít phương pháp phát hiện giấu tin trong video.

Steganography và steganalysis đang ở trong một trận chiến không bao giờ kết thúc. Bất cứ khi nào một kỹ thuật steganographic tốt được phát triển thì một kỹ thuật steganalysis cũng được phát triển và cố gắng để đánh bại nó [[12]](#_Tài_Liệu_Tham). Steganalysis là một kỹ thuật khoa học phát hiện các thông điệp bí mật ẩn bằng steganography [[13]](#_Tài_Liệu_Tham). Một khi có bằng chứng cho thấy một thông tin bị ẩn steganography mục tiêu sẽ bị đánh bại ngay cả khi tin nhắn không được trích xuất. Mặc dù kỹ thuật steganography có thể làm thông tin giấu trong suốt đối với mắt người, nhưng vẫn có thể tấn công chúng(Steganography). Bất kỳ quá trình nhúng nào chắc chắn sẽ để lại dấu vết trong đối tượng stego (đối tượng mang thông tin giấu) và làm thay đổi một số thuộc tính của nó. Đưa ra các đặc tính bất thường và một số sự suy giảm của chất lượng. Steganalysis phát hiện sự hiện diện hoặc vắng mặt của tin nhắn ẩn hoặc xác định thuật toán nhúng được sử dụng [[14]](#_Tài_Liệu_Tham). steganalysis hoạt động có thể thay đổi, trích xuất hoặc phá hủy các tin nhắn ẩn hoặc trích xuất một số thuộc tính của nó như chiều dài tin nhắn, … [[15]](#_Tài_Liệu_Tham).

Tồn tại rất nhiều phương pháp steganalysis khác nhau, một số phương pháp steganalysis cụ thể chỉ áp dụng cho những loại steganography nhất định. Có nghĩa là chúng áp dụng được với kỹ thuật steganography nào đó nhưng không hoạt động được với kỹ thuật khác[[16, 15]](#_Tài_Liệu_Tham). Trong khi đó những loại kỹ thuật steganalysis khác có thể phát hiện và phá hủy bất kỳ thông tin bí mật nào khác sử dụng kỹ thuật steganography. Ví dụ về phương pháp steganalysis cụ thể là một trong những đề xuất của Zaker et al. [[15]](#_Tài_Liệu_Tham) trong đó một phương pháp thống kê được thiết kế để ước lượng số bit bí mật ẩn bằng TPVD. Một phương pháp khác đã được giới thiệu bởi Su et al. [[16]](#_Tài_Liệu_Tham) là phát hiện thông tin ẩn bằng thuật toán steganography dựa trên vectơ chuyển động.

Phần này không đi sâu vào chi tiết các phương pháp steganalysic. Thay vào đó chúng ta chúng ta sẽ tìm ra phương pháp giúp thiết kế một kỹ thuật steganography mạnh mẽ. Vì vậy, chúng ta sẽ xem xét một cuộc tấn công steganalysis để đánh giá điểm mạnh và điểm yếu của thuật toán steganography được thiết kế. Trong thực tế, sau khi thiết kế một kỹ thuật steganography, nó cần được kiểm tra để biết điểm mạnh điểm yếu của nó. Kỹ thuật thiết kế nên được thứ nghiệm để chống lại một số cuộc tấn công xử lý hình ảnh, và cũng chống lại bất kỳ kỹ thuật steganalysis nào được biết đến nhằm chống lại các kỹ thuật được thiết kế. Ở đây kiểm tra các cuộc tấn công (steganalysis) được sử dụng thường xuyên trong sách để kiểm tra bất kỳ thuật toán steganography nào.

Phân tích rõ ràng sự tàng hình của thông tin giấu là yêu cầu đầu tiên và quan trọng nhất. Vì sức mạnh của Steganography nằm ở việc che giấu thông tin liên lạc. Vì vậy, thời điểm mắt người phát hiện một số thay đổi xảy ra với video thì lúc đó thuật toán thất bại. Loại thử nghiệm đầu tiên thực hiện bằng cách so sánh các khung hình video gốc và các khung hình video stego [[17, 18]](#_Tài_Liệu_Tham). Để đo lường tính vô hình trước tri giác con người, có rất nhiều số liệu có sẵn đã được đưa trong phần trước. Một trong những số liệu được sử dụng rộng rãi nhất là PSNR [[19, 20, 21, 22, 23, 17–18]](#_Tài_Liệu_Tham). Giá trị PSNR tối thiểu là 38 dB được chấp nhận làm thước đo chất lượng đối với hình ảnh khung hình stego [[24]](#_Tài_Liệu_Tham).

Phân tích biểu đồ là một cách đơn giản và hiệu quả khác để kiểm tra hiệu ứng thuật toán steganography trên các khung stego. Nó được thực hiện bằng cách so sánh biểu đồ của khung trước và sau khi nhúng để biết sự thay đổi về độ phân phối màu pixel do thuật toán steganography sử dụng. Thử nghiệm này được các tác giả sử dụng trong [[25, 26, 27, 24]](#_Tài_Liệu_Tham).

Tỷ lệ lỗi bit (BER) nó cho biết số lượng bit tin nhắn bí mật tồn tại và được trích xuất thành công từ video stego. BER được định nghĩa là tỷ lệ giữa số lượng các bit được trích xuất sai và tổng số bit tin nhắn ban đầu. Phương trình BER = Đếm số bit lỗi / Số lượng bit tin gốc. BER được sử dụng trong [[28, 29]](#_Tài_Liệu_Tham).

Độ bề khi nén: Vì một video không nén có kích thước rất lớn nên nó phải được nén trước khi truyền. Điều này dẫn đến chúng ta phải quan tâm đến việc kiểm tra độ bền của video stego khi bị nén. Điều này thường được thực hiện bằng cách nén các video stego bằng cách sử dụng các kỹ thuật nén khác nhau và với tốc độ nén khác nhau, để biết tỷ lệ sống sót của các bit ẩn. Tính linh hoạt đối với mã hóa MPEG-2 đã được các tác giả thử nghiệm trong [[17]](#_Tài_Liệu_Tham). Họ tính tỷ lệ giữa các bit được trích xuất chính xác và tổng số bit tin nhắn bí mật ở ba tốc độ bit khác nhau để biết tính linh hoạt của thuật toán của chúng đối với nén.

Độ bền để chống lại những thao tác với một video stego, có thể là cố ý hoặc vô tình gây ra những thay đổi của hình ảnh hoặc các hoạt động xử lý video. Ảnh hưởng của việc xử lý ảnh thông thường như dịch, xoay, cắt xén, làm mờ và đánh dấu sẽ được kiểm tra [[30]](#_Tài_Liệu_Tham). Các cuộc tấn công xử lý video phổ biến như thêm khung hình và giảm khung cũng nên được kiểm tra [[31]](#_Tài_Liệu_Tham). Báo cáo kết quả của các cuộc tấn công trên và các biện pháp cho một kỹ thuật steganographic nhất định cho một cái nhìn trung thực rõ ràng về những điểm mạnh và điểm yếu của phương pháp thiết kế.

## ***II- Phát hiện dựa theo đối tượng mang tin.***

### ***1. Phát hiện giấu tin dựa trên Exploring the Temporal Correlation between Frames (mối tương quan về thời gian giữa các khung hình).***

Budhia et. al [[1]](#_Tài_Liệu_Tham) đã đề xuất một kỹ thuật phát hiện giấu tin trong video bằng cách sử dụng thông tin dư thừa có trong miền thời gian như một cách chống lại các thông điệp bí mật được nhúng bởi kỹ thuật giấu tin trải phổ. Nghiên cứu dựa trên các phương pháp tích hợp tuyến tính của họ đã thành công trong việc xác định các dữ liệu được giấu tốn ít năng lượng với độ chính xác cao. Các kết quả mô phỏng cũng chứng minh ưu thế của các phương pháp dựa trên thời gian so với các phương pháp phát hiện dự liệu bí mật dựa trên không gian.

Sơ đồ 1 minh họa kỹ thuật giấu tin trong video và hệ thống phát hiện giấu tin được sử dụng trong [[1]](#_Tài_Liệu_Tham). Để bắt đầu, người gửi nhúng một dữ liệu bí mật dạng nhị phân vào video ban đầu để tạo ra video đã giấu tin giống với video ban đầu. Các bit dữ liệu bí mật được nhúng vào video ban đầu bằng cách điều chỉnh nó thành một tín hiệu được gọi là thủy vân số (Watermarking). Video đã giấu tin sau đó được truyền thông qua Internet tới người nhận. Sử dụng video đã giấu tin và khóa bí mật, người nhận sẽ trích xuất dữ liệu được giấu. Trên đường đến người nhận, dữ liệu có thể bị chặn bởi một người phát hiện giấu tin cảnh giác. Phát hiện thủy vân sẽ báo hiệu sự hiện diện của thông tin ẩn trong phương tiện. Mặc dù thủy vân được chèn vào một miền phi không gian (như DCT), nhưng nó được xác định trên cùng một miền với video ban đầu.

DỮ LIỆU NHÚNG

THUẬT TOÁN NHÚNG

KÊNH KẾT NỐI

PHÁT HIỆN GIẤU TIN (TRỰC TIẾP/GIÁN TIẾP)

THUẬT TOÁN TRÍCH XUẤT

DỮ LIỆU NHỊ PHÂN

DỮ LIỆU BÍ MẬT

DỮ LIỆU GIẢI MÃ

VỊ TRÍ GIÁM SÁT

KHÓA AN TOÀN

KHÓA AN TOÀN

NGƯỜI GỬI

NGƯỜI NHẬN

Sơ đồ 1: Giấu tin và Phát hiện giấu tin trong Video (Nguồn: [1])

Budia et. al đã đánh giá tầm quan trọng của việc khai thác mối tương quan thời gian trong phát hiện giấu tin trong video. Họ tạo ra một chuẩn(Sơ đồ 2) dựa trên phát hiện giấu tin của Gaussian Spread Spectrum (dựa trên các phương pháp các phương pháp giấu tin [[2, 3]](#_Tài_Liệu_Tham). Có hai giai đoạn thiết yếu: (i) Một giai đoạn tấn công Thủy vân để đánh giá phương tiện truyền thông ban đầu từ các phương tiện truyền thông đã giấu tin và (ii) Giai đoạn nhận dạng kiểu để phát hiện hoạt động giấu tin. Các thuật toán khác nhau có thể được thay thế cho mỗi giai đoạn để tạo ra các kỹ thuật phát hiện giấu tin cho một vài ứng dụng. Cách tiếp cận dựa trên giai đoạn cũng tạo điều kiện thuận lợi việc sử dụng các thuật toán nâng cao hiện nay cho các cuộc tấn công thủy vân và Nhận dạng kiểu.

CHUỖI VIDEO ĐẦU VÀO

TẤN CÔNG THỦY PHÂN

NGƯỜI PHÁT HIỆN VÀ MÁY DÒ

CÓ DỮ LIỆU ẨN HAY KHÔNG?

Sơ đồ 2: Nền tảng phát hiện giấu tin trong video (Nguồn:[1])

### ***2. Phát hiện giấu tin trong video dựa trên Asymptotic Relative Efficiency(ARE).***

Jainsky et. al [[4]](#_Tài_Liệu_Tham) đã đề xuất thuật toán phát hiện giấu tin trong video kết hợp với mối tương quan tiệm cận [[5]](#_Tài_Liệu_Tham). Thuật toán này phù hợp hơn cho các ứng dụng mà trong đó chỉ có một tập hợp con của các khung hình video được nhúng dữ liệu bí mật mà không phải tất cả chúng. Tín hiệu video giấu tin được giả định bao gồm một chuỗi các khung hình ảnh tương quan và tuân thủ mô hình tương quan thời gian Gauss-Markov. Phát hiện giấu tin bao gồm một giai đoạn phát hiện và sau đó là một giai đoạn xử lý tín hiệu. Các giai đoạn xử lý tín hiệu nhấn mạnh sự hiện diện của dữ liệu ẩn trong chuỗi các khung sử dụng chuyển động sơ đồ ước lượng. Giai đoạn phát hiện được dựa trên mối tương quan tiệm cận (ARE) [[5]](#_Tài_Liệu_Tham), trong đó cả video ban đầu và dữ liệu bí mật được nhúng được coi là biến ngẫu nhiên. Máy dò dựa trên ARE không có bộ nhớ trong tự nhiên và sử dụng ngưỡng thích ứng cho các đặc điểm của video mà được sử dụng để phân biệt một video ban đầu với video đã giấu tin. Các đặc điểm của video (ví dụ: kích thước, độ lệch chuẩn và tương quan hệ số) được coi là những đặc điểm sẽ thay đổi từ một chuỗi này so với một chuỗi khung khác. Phân tích chuỗi khung hình đi vào máy dò cũng coi là một tham số để phát hiện giấu tin.

### ***3. Phát hiện giấu tin trong video dựa trên Mode Detection.***

Su et. al [[6]](#_Tài_Liệu_Tham) đề xuất một thuật toán phát hiện giấu tin trong video nhằm phát hiện dữ liệu ẩn trong video được giấu tin bằng phần mềm Moscow State University (MSU), một trong số rất ít các công cụ giấu tin trong video có thể nhúng bất kỳ tập tin nào ở định dạng AVI (Audio Video Interleave) và các dữ liệu được nhúng có thể được trích xuất một cách chính xác ngay cả sau khi các video đã giấu tin được nén. Thuật toán phát hiện giấu tin sử dụng sự tương quan giữa các khung liền kề và tìm ra một chế độ phân phối đặc biêt trên các khung. Ví dụ đơn vị nhúng là khối 32 x 32 điểm ảnh và bốn khối 16 x 16 điểm ảnh, với mỗi đơn vị tạo thành mô hình phân phối giống như bàn cờ. Sau khi phân tích sự tương quan giữa các khung liền kề, nếu tỷ lệ của số khối 32 x 32 điểm ảnh so với chế độ phân phối đặc cho ra tổng số khối 32 x 32 điểm ảnh vượt trên ngưỡng chuẩn thì video tín hiệu được dự đoán sẽ mang một thông tin nhúng.

### ***4. Phát hiện giấu tin trong video dựa trên Spatial and Temporal Prediction (dự đoán trên miền không gian và thời gian).***

Pankajakshan và Ho đề xuất một chương trình phát hiện giấu tin trong video [**[7]**](#_Tài_Liệu_Tham) cho tiêu chuẩn mã hóa video MPEG trong đó một khung nhất định được dự đoán từ các khung tham chiếu lân cận sử dụng bù trừ chuyển động [[8]](#_Tài_Liệu_Tham). Lược đồ mã hóa MPEG hỗ trợ hai loại khung được dự đoán: các khung hình P (sử dụng một khung hình đơn lẻ làm khung tham chiếu) và khung hình B (sử dụng khung quá khứ và tương lai khung làm khung tham chiếu). Dự đoán khung lỗi (The prediction-error frames(PEF)) tương ứng với các khung P và B sau đó được mã hóa bằng kỹ thuật mã chuyển đổi. PEF thể hiện mối tương quan về thời gian và không gian giữa các khung liền kề. PEF của tín hiệu video thử nghiệm được phân tích bằng cách sử dụng phương pháp Discrete Wavelet Transform - DWT 3 cấp (Chuyển đổi sóng rời rạc) và ba vị trí đầu tiên của các hàm đặc trưng (characteristic functions-CF) để tính toán. Các vecto kết quả đặc trưng được đưa ra nhằm tạo ra cách phân loại để phân biệt các video được giấu tin và không được giấu tin.

### ***5. Các kỹ thuật phát hiện giấu tin khác***

Kancherla và Mukkamala [[9]](#_Tài_Liệu_Tham) đề xuất một phương pháp phát hiện giấu tin sử dụng mạng nơron và máy hỗ trợ vector(SVM) để phát hiện dữ liệu ẩn bằng cách xem xét miền không gian và thời gian dư thừa. Zhang et. al [[10]](#_Tài_Liệu_Tham) đề xuất một phương pháp phát hiện giấu tin trong video dựa trên kỹ thuật trải phổ. Mô hình của họ mô phỏng video ban đầu và dữ liệu ẩn độc lập và sử dụng hàm xác suất khối dựa trên tín hiệu khác biệt giữa các khung hình nhằm để lộ hiệu ứng răng cưa (biến dạng) gây ra bằng cách nhúng dữ liệu. Liu et. al [[11]](#_Tài_Liệu_Tham) đề xuất một sự tương quan khung dựa trên thuật toán phát hiện giấu tin trong video nén dựa trên sự tương quan khi sử dụng thông đồng để trích xuất các tính năng từ các khung video tương tự của một cảnh duy nhất và sử dụng một nguồn cấp dữ liệu chuyển tiếp mạng nơron có khả năng ánh xạ phi tuyến tính để phân loại mù.

## ***III- Phát hiện dựa theo thuật toán giấu tin***

### ***1. Tệp hình ảnh.***

* Dữ liệu ẩn trong một hình ảnh có thể được phát hiện bằng cách xác định các thay đổi về kích thước, định dạng tệp tin, thời gian sửa đổi cuối cùng và bảng màu của tệp tin. Các phương pháp phân tích thống kê có thể được sử dụng.
* Giả sử rằng bit ít quan trọng nhất ít nhiều ngẫu nhiên là giả định không chính xác vì áp dụng một bộ lọc cho thấy các LSB có thể tạo ra một hình ảnh dễ nhận biết. Do đó, có thể kết luận rằng LSB không phải là ngẫu nhiên. Đúng hơn, chúng bao gồm thông tin về toàn bộ hình ảnh.
* Khi một thông điệp bí mật được chèn vào một hình ảnh, các LSB không còn ngẫu nhiên. Với dữ liệu được mật mã có entropy cao, LSB của trang bìa sẽ không chứa thông tin về bản gốc và ít nhiều ngẫu nhiên. Bằng cách sử dụng phân tích thống kê trên LSB, sự khác biệt giữa giá trị ngẫu nhiên và giá trị thực có thể được xác định.

### ***2. Tập tin âm thanh.***

* Các phương pháp phân tích thống kê có thể được sử dụng cho các tập tin âm thanh kể từ khi sửa đổi LSB cũng được sử dụng trên âm thanh.
* Các kỹ thuật sau đây cũng hữu ích cho việc phát hiện dữ liệu ẩn:
* Quét thông tin cho các tần số không nghe được.
* Xác định các biến dạng lẻ và các mẫu hiển thị sự tồn tại của dữ liệu bí mật.

# **B. NGHIÊN CỨU: STEGANALYSIS - PHÁT HIỆN GIẤU TIN**

## ***I. Sơ lược về Steganalysis***

Quá trình phát hiện thông tin bí mật với xác suất cao và ít phức tạp thông qua các môi trường đa phương tiện vô hại được gọi là Staganalysis, tiếng việt là Phát hiện giấu tin. Nó còn được sử dụng dùng để xác định liệu 1 vật chứa có chứa thông tin mật hay không. Vật chứa ở đây là môi trường vô hại ví dụ như audio, video, văn bản, … Xác định phải được tự động hóa để có thể có hiệu quả và giám sát thực tiễn hoặc theo dõi các hành vi như vậy [3]. Do đó, cần phải có các phương pháp hiệu quả và đáng tin cậy để phát hiện ra sự hiện diện của dữ liệu bí mật được giấu trong môi trường vô hại.

Để thiết kế được một hệ thống phát hiện giấu tin tự động, chúng ta nên tìm kiếm thống kê các thay đổi mọi nơi trong môi trường mà được sử dụng để giấu tin. Các thay đổi có thể được định lượng và so sánh với ngưỡng nào đó hoặc cơ sở dữ liệu đã biết để đi đến quyết định là có giấu hay là không.

Một hệ thống giấu tin liên quan đến hai bên: người gửi nhúng các thông điệp bí mật vào đối tượng chứa và người nhận đã trích nó ra. Việc bảo mật một phần đến từ sự hiện diện của một khoá bí mật K trong hệ thống mà thể hiện chi tiết cách thông tin bí mật được nhúng và trích xuất. Chúng ta giả định rằng K được trao đổi an toàn giữa người gửi và người nhận trước khi truyền thông bí mật; chìa khóa này là cụ thể cho từng thuật toán giấu tin và có thể chứa thông tin ví dụ như là cách bảo mật và thông tin giấu ở đâu trong đối tượng chứa hay điểm khởi tạo cho bộ tạo số giả ngẫu nhiên. Trong cách xây dưng này, môi trường được sử dụng cho truyền thông bí mật là video kỹ thuật số.

Steganographic

Extraction

Algorithm

PUBLIC CHANNEL

Steganographic

Extraction

Algorithm

**Steganalysis**

Secret Key K

**Steganagraphy**

Secret Key K

Cover

Object

Binary

message

EAVESDROPPER OR MONITORING POINT

Decoded Secret Message

Is something hidden?

SENDER

RECEIVER

Sơ đồ 3: Mô hình Steganalysis trong thực tế

Người gửi sẽ lấy video , đại diện cho video chứa, sau đây gọi là “host video”, viết tắt là “host”, mà nhúng một vectơ thông điệp bí mật dưới dạng nhị phân sử dụng khóa K để tạo ra video có chứa tin giấu (gọi là stego-video, hay video watermarked, viết tắt watermarked) gần như video ban đầu. Các stego-video sau đó được truyền đạt dọc theo một kênh công cộng đến cho người nhận. Tại người nhận, stego-video và khóa bí mật K được sử dụng để trích xuất tin nhị phân bí mật. Mục tiêu của nhà thám mã giấu tin là giám sát kênh công cộng và phát hiện sự hiện diện của bất kỳ thông tin bí mật nào trong một chuỗi video nhất định.

Host video ban đầu của được biểu thị là Uk(m, n), trong đó 1 ≤ k ≤ N là số khung hình của video, và m, n là các chỉ số hàng và cột của các điểm ảnh tương ứng. Thông điệp bí mật nhị phân được nhúng vào video gốc bằng cách điều chế nó thành một tín hiệu(gọi là watermark) được biểu thị Wk(m,n). Để tương thích, watermark Wk(m, n) được định nghĩa trên cùng một miền với host video Uk(m, n). Tín hiệu stego-video sau khi nhúng tin được biểu diễn bằng phương trình được sử dụngphổ biến :

**Xk(m, n) = Uk(m, n) + αk(m, n) · Wk(m, n) , k = 1, 2, 3 . . . N , (1)**

trong đó αk(m, n) là một nhân tố tỷ lệ được sử dụng để điều chỉnh nồng độ của thông điệp ẩn để cân bằng giữa cảm giác nhìn và sức chịu đựng. Trong thực tế, để đơn giản α được coi là hằng số trên tất cả các điểm ảnh và khung. Do đó, công thức 1 được biểu diễn lại như sau:

**Xk(m, n) = Uk(m, n) + αk · Wk(m, n) , k = 1, 2, 3 . . . N , (2)**

Kết quả α · Wk(m, n), trên thực tế, là một hàm của thông điệp bí mật nhị phân, khóa bí mật K và host Uk(m, n). Mối quan hệ giữa các tham số này được quyết định bởi thuật toán nhúng. Nói chung, mỗi thuật toán steganographic có thể được biểu diễn bằng phương trình (2). Để có một tham chiếu thích hợp cho việc phát hiện giấu tin có hiệu quả, chúng ta phải đưa ra một số giả định về phương pháp nhúng như đã thảo luận trong phần tiếp theo.

## ***II. Đặt vấn đề***

Tổng thể mục tiêu là thiết kế 1 phương pháp phát hiện giấu tin cho các video số, mà tối ưu hơn so với đề xuất trước đây của phương pháp hình ảnh là theo từng khung hình mà không tính đến sự dư thừa thời gian để có phát hiện chính xác cao hơn. Chúng ta sẽ xem xét vấn đề này đầu tiên bằng cách giới hạn quá trình xử lý video vào miền thời gian. Các phương pháp hình ảnh hoạt động trên miền không gian trực giao sau đó có thể dễ dàng phối hợp để nâng cao hiệu xuất so với các phương pháp trước đấy. Chúng ta sẽ tập trung vào phát hiện giấu tin mà phương pháp giấu dựa trên trải phổ do tính phổ biến và ảnh hưởng của nó trong nghiên cứu.

Bản chất, chúng ta cần xây dựng 1 hộp phán xét (decision box) mà đầu vào là 1 luồng video kỹ thuật số, và kết quả là có hay không có thông tin được giấu, bằng cách sử dụng các thông tin về thuật toán giấu và các mẫu về dư thừa thời gian trong khung hình video kỹ thuật số. Không cần biết về khóa K. Cụ thể, chúng em giả sử rằng kỹ thuật giấu tin dựa trên spread spectrum bằng cách thêm Gaussian watermark vào vùng không gian hoặc tần số của mỗi Frame. Do đó, những giả định sau đây là cần thiết. Đầu tiên, chúng em yêu cầu thông tin được nhúng vào mỗi Frame Wk là độc lập, không có ý nghĩa và là phân phối chuẩn Gaussian. Thứ 2, người gửi nhúng những thông tin vào mỗi pixel của mỗi frame của video sequence, giả định này là hợp lệ bởi vì để tối ưu hóa khả năng giấu tin, người gửi cần làm cho sử dụng càng nhiều tín hiệu gốc càng tốt để tối ưu hóa khả năng giấu thông tin. Tuy nhiên, có 1 sự cân bằng giữa bảo mật giấu tin và khả năng truyền tải sẽ được trình bày sau này.

Hình sau hiển thị kết quả giấu tin cho một khung hình đơn để làm sáng tỏ khái niệm. Hình 1(a) là khung lưu trữ còn được gọi là đối tượng chứa hoặc khung chứa, và hình 1(b) là đối tượng stego hoặc stego-video frame chứa dấu thủy chuẩn Gaussian (được khuếch đại cho cảm giác thị giác) thể hiện trong hình 1(c) với α = 5.



Hình 1: Giấu tin trong 1 ảnh tĩnh sử dụng Spread Spectrum

Các số liệu được sử dụng để đánh giá thành công của thuật toán là xác xuất giữa phát hiện nhầm (false-positive) và có giấu nhưng không phát hiện được(false-negative) được định nghĩa như sau. Xác suất của phát hiện nhầm là khả năng phát hiện mà có thông tin được giấu trong video nhưng thực tế không có cái gì được nhúng vào video (α = 0);Xác suất phát hiện sai là khả năng phát hiện mà không có thông tin được giấu nhưng thực tế là có thông tin được giấu (α khác 0), tức là không phát hiện được. 1 kỹ thuật giấu tin tốt nên cố gắng để giảm khả năng phát hiện lỗi. Tuy nhiên, đối với an ninh mạng hay các ứng dụng tin học máy tính, điều bắt buộc là tỷ lệ phát hiện false-negative phải thấp hơn. Do đó, bỏ qua các trường hợp phát hiện false-positive đối với phát hiện false-negative có thể là cần thiết thông qua việc lựa chọn các ngưỡng thuật toán thích hợp. Xử lý tiếp trên tín hiệu video được đánh dấu bằng kỹ thuật của chúng em để có thể tùy biến để có kết quả chính xác hơn.

**Steganalysis Decision Box**

**No hidden information present**

**Hidden information present**

**OR**

**Partial information and assumptions about potential Embed Method**

**Degital Video Sequence under consideration**

Sơ đồ 4: Cơ bản phát hiện giấu tin video

## ***III. Xây dựng thuật toán***

### ***1. Thuật toán***

Phương hướng của hầu hết các phương pháp phát hiện giấu tin là tạo tìm ra các đăc trưng mà có phân biệt giữa các đặc tính chung giữa có nhúng và không có nhúng. Đặc trưng này thường được so sánh 1 cách ngầm định hoặc rõ ràng đến một ngưỡng nào đó để quyết định có hay không một tín hiệu Yk(m, n) có chứa thông tin ẩn hay không. Do đó, nhiều nghiên cứu về phát hiện giấu tin trên hình ảnh, đã tập trung vào việc xác định các tính năng hình ảnh mà thay đổi khi áp dụng các thuật toán giấu tin. Các nhà nghiên cứu thường sử dụng công nghệ xử lý hình ảnh và bộ công cụ thống kê mà trong một số hình thức cố gắng để ước tính 1 vât chứa tiềm năng **Ûk(m, n) = H[Yk(m, n)]** từ Yk(m, n). Vật chứa ước tính Ûk(m, n) này sau đó bằng cách nào đó đươc so sánh với Yk(m, n) theo thứ tự để phát hiện nếu thứ gì đó được giấu. Giả định cơ bản rằng độ lệch của các đặc trưng riêng Yk(m, n) và Ûk(m, n) sẽ khác nếu có gì đó được nhúng vào (i.e., Yk(m, n) = Xk(m, n) = Uk(m, n) + α·Wk(m, n)) so với khi không có gì được nhúng vào Yk(m, n) (i.e., Yk(m, n) = Uk(m, n)). Độ lệch này sẽ được trích xuất các thông tin cần thiết áp dụng trong việc phân loại sau này.

Ở đây, chúng ta xây dựng một hướng đi mới cho vấn đề này sử dụng các nghiên cứu trước đây về các cuộc tấn công trên thủy phân số. Ưu điểm là thay vì sử dụng các thư viện xử lý hình ảnh và chức năng thống kê để xác định các ứng cử viên tiềm năng cho việc phân tích phát hiện giấu tin. Chúng tôi mượn các nghiên cứu trong các lĩnh vực liên quan đến thủy vân số. Hơn nữa, cách tiếp cận này là chung chung và có thể làm mục tiêu để xác định các loại cụ thể của giấu tin bằng cách thay thế khối chung với các thuật toán thích hợp.

Hình sau mô tả cách hoạt động của thuât toán này. Video được xem xét Yk(m, n) (được tách thành N frame tạo thành 1 chuỗi) được đưa vào một khối tấn công thủy phân số nhằm ước tính host signal để tính ra ước lượng Ûk(m, n). Khối này có thể phỏng đoán sự hiểu biết thuật toán nhúng có hiệu quả. Ước lượng của watermark Ŵk, được tính bằng cách tính độ lệch giữa Yk(m, n) và Ûk(m, n), sau đó sẽ được đưa vào bộ phân loại mẫu. Nếu một cái gì đó thực tế được nhúng trong Yk(m, n), thì đầu vào của bộ phân loại sẽ là 1 tín hiệu Gaussian watermark bị sửa đổi bởi nhiễu do lọc (tấn công watermarking). Nếu không có gì được nhúng trong Yk(m, n) thì Ŵk chỉ đơn gian là các nhiễu do bộ lọc. Trong trường hợp lý tưởng, nếu bộ lọc không có nhiễu, ước tính Ŵk sẽ bao gồm watermark Gaussian ban đầu được nhúng trong trường hợp video được giấu và sẽ bằng 0 cho video không được giấu. Bằng cách sử dụng một số thông tin về thuật toán nhúng, sự phân biệt giữa hai trường hợp này có thể được thực hiện để phát hiện sự có mặt của thông tin bí mật.

**No hidden information present**

**Hidden information present**

**OR**

**Degital Video Sequence under consideration**

**Attack**

**Difference**

**Pattern Classifier**

Partial information and assumptions about potential Embed Method

Steganalysis

Decision Box

Sơ đồ 5: Thuật toán Steganalysis sử dụng Collusion Attack

Quy lại, chúng ta sẽ tập trung vào các mối tương quan thời gian giữa các khung hình video để ước tính một video gốc (ở đây chúng ta sẽ sử dụng Collusion attack). Và sẽ sử dụng mô hình phân loại hiệu quả, sử dụng các đặc trưng là nếu có giấu tin, kỳ vọng bằng 0 và là 1 phân phối chuẩn Gaussian. Phần tiếp theo sẽ giới thiệu cách sử dụng collussion attack để ước lượng video gốc từ tín hiệu hận được Yk(m, n), chính là khối ‘Attack’ trong sơ đồ trên. Collusion attack là chìa khóa của thuật toán này.

### ***2. Collusion attack (Tấn công tích hợp)***

Tích hợp, tiếng anh là collusion, cho kỹ thuật chụp ảnh số và steganography đề cập đến việc sử dụng nhiều khung hình ảnh (có thể hoặc không thể tạo thành một chuỗi video) để loại bỏ sự hiện diện của một watermark trong một hoặc nhiều khung hình ảnh. Nhìn chung, collusion attack có thể là tuyến tính hoặc phi tuyến tính để khai thác sự sai khác và sự tương đồng giữa các frame để thận trọng làm giảm năng lượng của watermark thông tin gốc. Chúng tôi đại diện cho “collusion” của một chuỗi các khung hình video để tạo ra một khung kết quả có nội dung watermark thấp hơn như sau:

**(3)**

Trong đó là kết quả cuả quá trình collusion attack ( gọi là “colluded”) và ở đây nó đại diện cho frame ước tính thứ k của video gốc Uk của host video và C là toán tử collusion mà khai thác sự tương quan và sự sai khác tất cả hoặc một tập hợp con chọn lọc của các khung ảnh có watermark X1, X2, . . . , XN để tính toán . Nhìn chung kết quả colluded sẽ chứa ít thông tin của watermark Wi(m, n) hơn khi so với Xi(m,n). Các dạng phổ biến của toán tử colluded Ci bao gồm việc lấy tối đa, nhỏ nhất, trung bình hoặc trung vị theo pixel-by-pixel trên một loạt khung hình ảnh.

Tích hợp tuyến tính là một trường hợp đặc biệt, trong đó C đại diện cho một hoạt động lấy trung bình trọng số của khung video được lựa chọn. Một cách trực quan, sự tích hợp tuyến tính trên một chuỗi các khung hình video khuếch đại các bộ phận của các khung hình tương tự và làm giảm các thành phần khác nhau. Vì vậy, nếu mối tương quan tuyến tính giữa các frame của video gốc Ui khác với mối tương quan giữa các frame watermark Wi trên cùng 1 vùng i như nhau thì tích hợp tuyến tính sẽ thành công trong việc làm giảm hoặc khuếch đại sự có mặt của watermark trong frame kết quả **.**

**Chú ý:** Chúng ta sẽ chỉ tập trung vào việc áp dụng cho trường hợp giấu tin bằng trải phổ trên video mà thực tế cho thấy rằng Wk(m, n) là độc lập cho mỗi khung. Chúng ta giả định rằng chuyển động trong video "chậm" có nghĩa là các khung hình liền kề tương tự nhau. Bởi vì mối tuơng quan thị giác này, trên 1 dãy k phần từ trung tâm ở i, các khung hình watermarked có thể được tính trung bình để bớt sự hiện diện của watermark trong frame thứ i.

Giả sử rằng chúng ta sử dụng một cửa sổ trượt để biểu thị khu vực thời gian được sử dụng cho tính trung bình khung; cửa sổ này được cho là có các khung hình tương tự. Cụ thể, chúng ta lấy một cửa sổ trượt kích thước 2L + 1 frame mà trung tâm là khung thứ k (trừ phần đầu và cuối của chuỗi frame kể từ khi cửa sổ đi ra ngoài dải i) để tính trung bình chuỗi video. Vì vậy, ước lượng của frame gốc thứ k theo toán tử lấy trung bìnhcho tích hợp tuyến tính được biểu diễn như sau:

trong đó Xi ~ Yi là các frame nhận được từ việc tách video bắt được trên đường truyền.

Tổng hợp với mọi i chạy từ đầu đến cuối chuỗi frame của video (bao gồm cả phần đầu và cuối của chuỗi frame), như sau:

(4)

trong đó, k là khung được xem xét để tính , một ước tính Uk. Tiếp theo chúng ta sẽ chứng minh tại sao chúng ta khẳng định rằng ≈ Uk .

Thay Xi = Ui + α.Wi cho tất cả k từ công thức (2) vào công thức (4) chúng ta thu được:

, (5)

trong đó, chúng ta tổng hợp tất các các khoảng của k và để biến i chạy trong khoảng đó.

Vì các watermark Wk(m, n) là độc lập và có kỳ vọng bằng 0, nên toán hạng thứ 2 của vế phải phương trình (5) tiếp cận 0 khi L đạt đến 1 giá trị nào đó. Hơn nữa, vì chúng ta giả sử Uk ≈ Ui cho tất cả i trong vùng lân cận của cửa sổ trượt tâm ở k, vì thế dẫn đến xấp xỉ sau:

Độ hiệu quả của (xấp xỉ Uk) phụ thuộc vào giá trị của L liên quan đến tốc độ chuyển động trong video. Thực nghiệm và nghiên cứu chỉ ra rằng với một giá trị tối ưu của L sẽ đảm bảo watermark sẽ bị loại bỏ bớt và đảm bảo cho giả đinh Ui ≈ Uk là chính xác.

Nếu tích hợp được áp dụng cho một chuỗi video nhất định Yk mà có thể có hoặc không có chứa watermark, chúng tôi tin rằng trong cả hai trường hợp đối với video chuyển động chậm và một giá trị được chọn lọc thích hợp của L, việc tìm xấp xỉ của Ui sẽ có hiệu quả . Vì vậy, nếu watermark được nhúng trong video, trừ Yk cho , ta được độ lệch Yk - ≈ Yk - Uk = αWk, là một ước lượng tỉ lệ của thủy vân Gaussian( được gọi là scaled watermark, có thể hiểu là watermark được co giãn theo 1 tỉ lệ nào đó) có kỳ vọng về 0. Và nếu không có watermark có trong Yk thì kết quả sẽ độc lập với bất kỳ đặc điểm nào của Gaussian mà chúng ta giả định từ trước cho watermark. Độ lêch này sẽ được sủ dụng trong 1 bộ phân loại đươc nói đến ở phần tiếp theo.

Trong trường hợp video không có nhúng tin bí mật, chúng ta có Yk = Xk = Uk + α.Wk , mà α = 0. Khi đó, ước lượng của scaled watermark sẽ là:

Ŵk = Yk −

Ŵk = Yk − CL(X**k**)

Ŵk = Uk + α.Wk - CL( Uk + α.Wk )

Ŵk = Uk - CL( Uk ) (do α = 0)

Ŵk = nk

trong đó, nk = Uk - CL( Uk )

(CL là phép toán lấy trung bình trong tích hợp tuyến tính được nói ở trước )

nk là "nhiễu" dư ra từ quá trình tích hợp tuyến tính và là thước đo sự bất biến của toán tử tích hợp C áp dụng cho dữ liệu đầu vào không phải là 1 video có giấu tin là hợp lệ.

Lý tưởng nhất là CL(Uk) ≈ Uk.

Trong trường hợp có giấu tin, Yk = Xk = Uk + α.Wk .Khi đó, ước lượng của scaled watermark sẽ là :

Ŵk = Yk −

Ŵk = Yk − CL(X**k**)

Ŵk = Uk + α.Wk - CL( Uk + α.Wk )

Ŵk = Uk – CL(Uk) + αWk - CL( α.Wk )

(Vì CL(a + b) = CL(a) + CL(b) trong trường hợp tích hợp tuyến tính. )

Ŵk = nk + W’k

(trong đó W’k = α(W k − C(Wk) ) )

Trong trường hợp này thì ước lượng của watermark là tổng của nhiễu do tấn công tích hợp và một tín hiệu Gaussian có một mối liên hê rất chặt chẽ với watermark nhúng Wk. Trong trường hợp tất cả các frame gốc đều giống nhau, nk sẽ bằng 0 và ước tính của watermark Ŵk sẽ là watermark nhúng Wk.

**\*Giả thiết toán học**

Vấn đề trong phát hiện giấu tin có thể được phát biểu bằng toán học là bài toán kiểm định các giả thiết.

Giả thiết H0: Ŵk = nk k = 1,2,…,N nếu không giấu tin mật.

Giả thiết H1: Ŵk = nk + W’k  k = 1,2,…,N nếu có giấu tin mật.

trong đó, nk là nhiễu còn sót lại như đã nói ở trên và W’k là 1 tín hiệu watermark Gaussian.

Mục đích của phát hiện giấu tin là để phân biệt giữa hai tình huống và đồng thời giảm thiểu xác suất của dương tính giả (false positive) và âm tính giả (false negative).

+ Xác suất của âm tính giả chính là xác xuất chọn H0 nhưng thực tế kết quả là H1.

+ Tương tự vậy, xác suất của dương tính giả chính là xác xuất chọn H1 nhưng thực tế kết quả là H0.

**\*Lý luận và phân tích lý thuyết**

Phương pháp steganalysis sẽ đưa ra các đặc trưng của watermark và sử dụng bộ nhận dạng mẫu để tìm kiếm các thông điệp ẩn. Do đó độ chính xác của các watermark ước tính sẽ liên quan đến độ chính xác của các giả thiết kiểm định.

Trong phần này, chúng ta sẽ nghiên cứu hiệu suất của việc tích hợp, giới hạn của ước lượng về các tham số nhúng sẽ dẫn đến thất bại của việc phát hiện giấu tin dựa trên tấn công tích h và tìm chiều dài tối ưu cho tấn công tích hợp.

Sẽ có các giả thiết sau cho chuỗi video và watermark frame.

(1) Frame gốc Uk là các phân phối mà có kỳ vọng là μ và phương sai là

(2) Mô hình tương quan của các frame gốc biểu diễn theo mô hình Markov thứ nhất mà trong đó tương quan giữa khung Ui và Uj được cho bởi ρ| i − j |  Ở đây ρ là hệ số quan hệ giữa hai khung liền kề.

(3) Các watermark frame Wk được giả định là độc lập với Uk và với các watermark frame khác, và là một biến đổi của phân bố Gaussian có kỳ vọng là 0 và phương sai . Vì watermark được nhúng với cường độ nhúng α nên phương sai watermark là .

Đối với video chuyển động chậm, tức là bối cảnh cảnh thay đổi không rõ ràng, chúng ta có thể hợp lý hóa giả thiết rằng các khung có cùng giá trị kỳ vọng và phương sai như đã nêu trong giả thiết (1). Trong giả thiết (2), chúng ta đòi hỏi sử dụng mô hình Markov thứ nhât để mô hình hóa mối tương quan giữa các frame trong một video. Theo trực giác, chúng ta thấy rằng sự tương quan giữa một khung tham chiếu và các khung khác là giảm đi khi trong một trong số chúng chuyển động cách xa khung tham chiếu. Mô hình này làm giảm sự tương quan sử dụng ρ| i − j |, trong đó|i − j| là khoảng cách giữa khung tham chiếu và các khung khác trong các giới hạn về chỉ số của của các khung. Chú ý là ρ| i − j | giảm khi khoảng cách tăng và

**\*Hiệu quả của Linear Collusion Attack**

Để đánh giá độ hiệu quả này, chúng ta sử dụng Mean Squared Error(MSE) giữa ước tính của watermark Ŵk và watermark nhúng αWk. MSE của 1 phép ước lượng là trung bình của bình phương các sai số. Để có được ước tính tốt nhất, MSE dự kiến sẽ phải là nhỏ nhất có thể. MSE cho mỗi Frame được tính bằng công thức dưới đây:

Phương trình này cho thấy giá trị của MSE giữa watermark ước tính và watermark ban đầu giống như giá trị kỳ vọng của MSE giữa frame ban đầu và frame được ước lượng bằng collusion.

**Mệnh đề 1:** Cho 1 chuỗi các khung frame video Xk, k=1,2..,N như đã trình bày trong công thức 2, theo các giả thiết (1), (2) và (3), MSE giữa watermark ban đầu và watermark ước tính thu được từ Collusion Attack được cho bởi,

trong đó, z = 2L+1

Chứng minh: xem phụ lục.

Tiếp theo, chúng ta giới thiệu khái niệm về no-collusion (không dùng phép tích hợp). Non-collusion là các tấn công tích hợp mà số lượng các khung hình colluded là một, z = 1. Trong trường hợp bình thường của non collusion, ước tính của watermark sẽ luôn bằng 0 bất kể nó là một watermarked hay không có watermark.

**Mệnh đề 2:** theo các giả thiết (1), (2) và (3), MSE giữa watermark ban đầu và watermark ước tính của một no-collusion được cho bởi công thức sau:

Chứng minh: xem phụ lục.

Vì ước tính của watermark luôn luôn là 0 trong trường hợp no-collusion, MSE giữa các watermarks luôn luôn bằng với phương sai của watermark được nhúng vào .

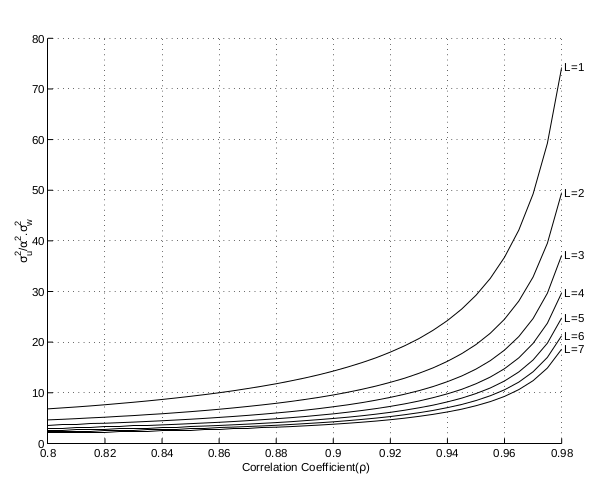
Tiếp theo sẽ phân tích sự thánh công của collusion attack. Để ý đến tỉ lệ của phương sai của watermark và phương sai của frame gốc. Đây là 1 độ đo SNR (signal-to-noise ratio), tỉ lệ tín hiệu trên nhiễu) trong đó watermark giao thoa với tín hiệu của frame gốc đại diện cho tín hiệu.

**Mệnh đề 3:** Từ mệnh đề 1 và 2, chúng ta thu được 1 giới hạn về tỉ lệ phương sai của frame gốc và phương sai của watermark được nhúng .

, trong đó, z = 2L + 1.

Chứng minh: xem phụ lục.

Chúng ta đặt 1 giới hạn của tỷ lệ phương sai của frame gốc và phương sai của watermark bằng cách đặt một ràng buộc cho MSE giữa các watermark trong linear collusion sẽ nhỏ hơn MSE của no-collusion. Chúng ta sẽ tìm ra các điều kiện để phương án sử dụng linear collusion thành công trong việc làm triệt tiêu watermark.

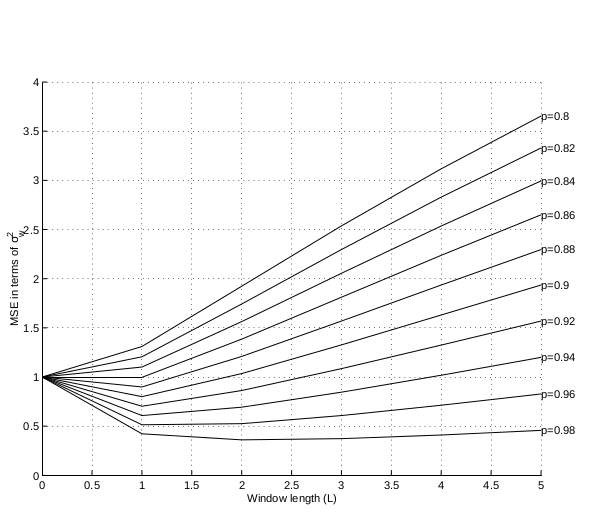
  
Figure 1. Giới hạn trên của tỉ lệ

Hình trên mô tả giới hạn trên của tỉ lệ phương sai của frame gốc và cường độ watermark với các giá trị của L và hệ số quan hệ ρ như phương trình đề cập trong mệnh đề 3. Chú ý z = 2L+1. Từ đây ta thấy rằng với ρ = 0.94 và L = 4 thì tỉ lệ này đạt giá trị lớn nhất và bằng 10. Điều này có nghĩa là phương sai của frame lớn hơn gấp 10 lần phương sai của cường độ watermark được nhúng vào với L = 4 sẽ sinh ra MSE cao hơn khi so với MSE khi không sử dụng collusion. Vì vậy chúng ta phải sử dụng các giá trị của L thấp hơn.

Lựa chọn L cao hơn giúp triệt tiêu watermark nhiều hơn ( vì ) . Tuy nhiên nếu như các frame là không tương tự nhau (ít tương quan với nhau) thì khi L tăng cũng sẽ làm tăng nhiễu trong quá trình thực hiện collusion attack. Vì vậy, buộc phải sử dụng 1 giá trị hợp lý của L để cân bằng giữa nhiễu và watermark triệt tiêu. và L cũng thỏa mãn để sai số MSE thấp hơn khi so với sai số MSE khi không sử dụng collusion.

Khi mà hệ số tương quan ρ tăng thì giới hạn của tỉ lệ cũng sẽ tăng theo cấp số mũ. Điều này nghĩa là đối với phương sai của 1 frame cố định thì khả năng để “tích hợp cho 1 video watermarked với cường độ nhúng α thấp” giảm đi khi mối tương quan giảm đi. Điều này cho thấy rằng khi cường độ watermark là nhỏ khi so với cường độ video gốc , vùng giới hạn cho các tham số sử dụng nằm ở phía tay phải hình trên. Nhưng đây chưa phải là tối ưu.

Trong hình dưới đây, giá trị MSE được biểu diễn dưới dạng là 1 hàm của L áp dụng với các giá trị của ρ với 1 số điều kiện của . đó là phương sai của frame gốc lớn hơn 10 lần so phương sai watermark . Với SRN = 0.1, ta thấy rằng nếu hệ số ρ > 0.86 thì ta sẽ có các điểm cực tiểu. Giá trị của L tương ứng với điểm cực tiểu sẽ là giá trị tối ưu của L và được sử dụng trong collusion attack với SRN và hệ số tương quan ρ cho trước. Theo quan sát chúng ta thấy rằng hệ số tương quan tăng sẽ làm tăng MSE.

  
Figure 2: Biểu đồ sai số MSE

Từ đây, ta thấy rằng giá trị tối ưu là L = 1 hoặc 2 ứng với ρ = 0.94 và 0.98.

Trước đây, chúng ta luôn giả sử rằng là các khung hình trong video phải tương tự nhau. Vì vậy nếu các khung hình trong video là không tương tự nhau thì việc xấp xỉ Uk sẽ gặp vấn đề. Để giải quyết nó, chúng ta sẽ sử dụng phương án tích hợp trọng số (Weighted Collusion Scheme ) và tấn công tích hợp dựa trên khối (block based collusion attack) được nhắc đến trong tài liệu đính kèm.

### ***3. Pattern Classifier***

Pattern Classifier (Bộ phân loại mẫu) giúp gán nhãn lớp cho các đối tượng từ một trong lớp trong dữ liệu đào tạo. Và ở báo cáo này, bộ phân loại này phải gán nhãn được cho 1 video có giấu và không có giấu tin, với đầu vào là 1 ước tính cho mỗi frame. Có 2 thành phần chính trong Pattern Classifier là trích xuất đặc trưng (Feature Extraction) và bộ phân loại (Pattern Classifier).

**3.1. Trích xuất đặc trưng (Feature Extraction)**

Trích xuất đặc trưng là 1 quá trình trích xuất ra các đặc trưng phân biệt từ 1 tập dữ liệu để phẫn biệt được là thuộc lớp nhãn (label ) nào. Ở đây chúng ta chỉ có 2 label là có giấu (1) và không có giấu (0). Các trích xuất được trích ra từ ước lượng của watermark sẽ hỗ trợ bộ phân loại trong việc phát hiện sự hiện diện dữ liệu bí mật trong chuỗi video hoặc giúp loại bỏ một trong hai giả thuyết đã nói ở phần trước.

Vì chúng ta giả thiết rằng việc giấu tin thực hiện bằng cách thêm watermark Gaussian. Chúng ta sẽ dụng các tính năng mà có thể đo lường mức độ Gaussianity trong một tín hiệu. Chúng bao gồm kurtosis, entropy và 25th percentile.

**a. Kurtosis**

Kurtosis là một giá trị mà trong đó một phần dùng để đo lường kiểu của phân phối. Kurtosis cho 1 phân phối Gaussian là 3 và cho hầu hết các phân phối khác là nhỏ hơn hoặc lớn hơn 3 phu thuộc vào loại phân phối. Kurtosis của 1 tập {x} cho trước được định nghĩa như sau,

Trong đó σ và μ đại diện phương sai và kỳ vọng của phân bố. Nó cũng dùng để tìm đỉnh của 1 phân bố. Tức là nếu nó có giá trị cao hơn thì nó sẽ có đỉnh cao hơn các phân phối cao hơn. Giá trị mong đợi cho Kurtosis của ước lượng là sẽ gần về 3. Trong trường hợp không có giấu thì đỉnh của ước lượng sẽ cao hơn.

Nhưng ở đây, kết quả từ collusion attack là 1 image với kích thước mxn. Nên chúng ta sẽ extract thông tin dựa trên biểu đồ của image thu được [2]. Như sau:

trong đó,

Average (μ): trung bình điểm ảnh, hay còn gọi là kỳ vọng

P: xác xuất.

Zk là giá trị pixel của điểm ảnh đó (0...255), biểu thị màu tại điểm ảnh đó.

nk là số lần mà điểm ảnh đó xuất hiện trong ảnh.

m\*n là kích thước của ảnh (dài x rộng)

var là phương sai toán học

σ là độ lệch chuẩn

Chú ý: k = 0 ... 255 vì giá trị ảnh không thể vươt quá vùng này. Áp dụng cho grey scale video. Trong trường hợp là video màu thì cần xử lý trước khi sử dụng.

**b. Entropy**

Entropy giúp xác định mức độ "ngẫu nhiên" trong một phân bố nhất định. Với phương sai cố định, phân bố Gauss có entropy lớn nhất. Do đó, Entropy tính được từ ước lượng của 1 video có giấu tin sẽ cao hơn entropy tính được từ ước lượng của 1 video không có giấu vì nhiều điểm hội tụ về không. Entropy được tính như sau,

,

trong đó, p(x) là ước lượng của phân phối Ŵk. Trong [38] các tác giả xác định thuật toán steganographic tốt là thuật toán có thể giảm thiểu sự gia tăng entropy do nhúng.

Theo các nghiên cứu toán học thì Entropy cho 2 trường hợp có giấu và ko có giấu tin trong video là khác nhau và vì thế là 1 đặc trưng tốt cho bộ phân loại. E0 đại diện cho Entropy của ước lượng từ video không có nhúng và đặt E1 là đại diện cho Entropy của ước lượng từ video có nhúng. Ước lượng của watermark chứa nhiễu dư thừa nk (do collusion attack) và tín hiệu Gaussian W’k như đề cập ở công thức trong phần Linear Collusion Attack. W’k là độc lập với nk và vì thế E1 có thể hiểu là tổng của E0 và Entropy của W’k

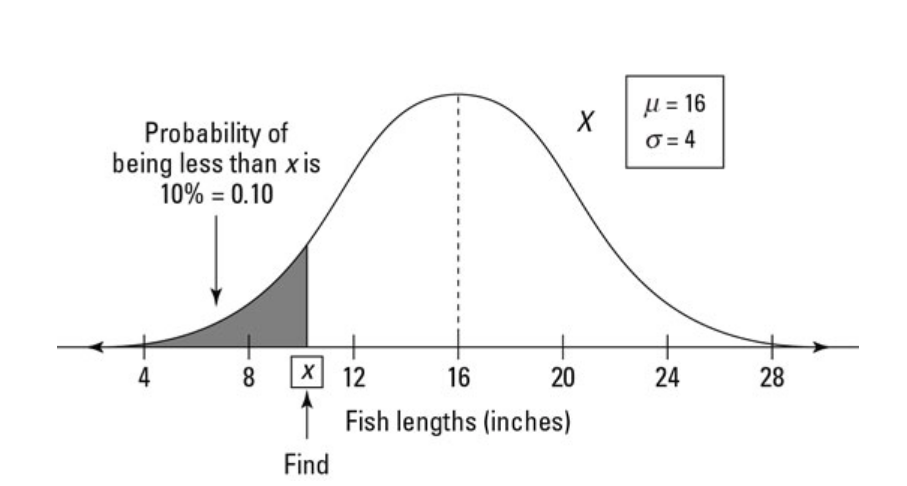
Mệnh đề 4. Entropy (E1) của ước tính của watermark thu được từ một video có giấu tin lớn hơn entropy (E0) của ước tính của watermark thu được từ một video không được giấu sử dụng Linear Collusion Attack.

trong đó,

Vì thế, khi L tăng độ lệch E1 và E0  càng tăng. Nếu độ lệch này đạt tối đa thì đây chinh là 1 đặc trưng sử dụng trong bộ phân loại. Đồng thời, tăng L sẽ triệt tiêu càng nhiều watermark, nhưng cũng sinh ra càng nhiều nhiễu, và cũng sẽ làm tăng MSE giữa các watermark. Vì vậy cần lấy 1 giá trị tối ưu của L để đủ phân biệt cũng như giữ nhiễu ở mức thấp.

**c. 25 percentile**

25th percentile là 1 giá trị x mà tại đó, nhiều nhất là 25% số điểm trên biểu đồ thấp hơn giá trị x này. Hay tìm x để p(X < x) = 0.25. Ta gọi giá trị này là x25.

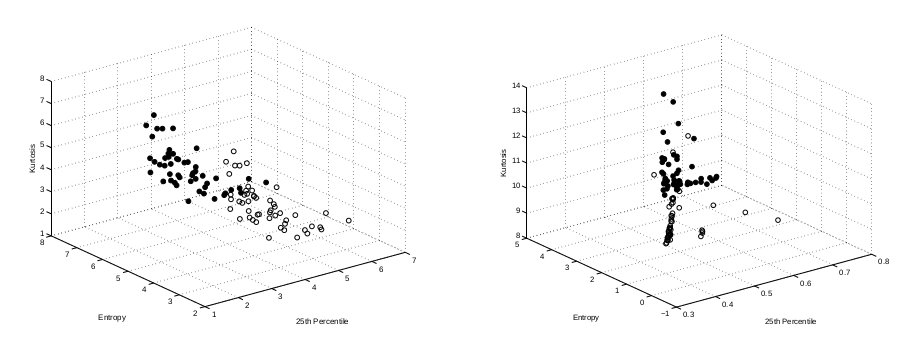


Cách tính là tra bảng z-table (phụ lục), ta tìm được z25 = 0.67.

**x25 = µ + σ.z25,**

trong đó, µ va σ là kỳ vọng và độ lệch chuẩn của phân phối.

Như vậy, chúng ta đã tính toán được đặc trưng cần tính toán sử dụng trong kNN.



B

iêB

Biểu đồ biểu thị vector (Kurtosis, Entropy, 25Percentile) trong 2 video test, mỗi video trong 2 trường hợp có giấu tin và không?

Màu đen: Frame trong video có giấu tin

Màu trắng: Frame trong video có giấu tin

**3.2. Pattern Classifier**

Bộ phân loại chúng ta sử dụng là kNN Classifier. Thuật toán này sử dụng các vector đặc trưng trích xuất từ test data và so sánh tìm các data tương tự trong trainning set để gán label cho data test. Cụ thể, nó(test data) sẽ tìm k điểm gần nó (trainning data) nhất để tính toán label của nó. Ở đây chúng ta sẽ dùng khoảng cách Euclid để tìm k điểm gần nhất. Và trong số k điểm gần nhất, nó sẽ lọc ra label nào được dùng nhiều nhất thì nó sẽ chọn label ấy làm label của mình. Input của thuật toán bao gồm sô k, cách xác định 2 điểm được xem là gần nhau, và trainning set. Giá trị của k đươc tính toán qua thực nghiệm, còn trainning set được chọn sử dụng Cross Vadilation.

Khoảng cách Euclid giữa 2 điểm p(p1, p2, ..., pn), q(q1, q2, ..., qn) (n = 3 trong bài toán này):

\*Trainning

Chúng ta sẽ chọn ra 14 chuỗi video bao gồm các video có đặc trưng khác nhau mà đại biểu cho 1 thể loại video. Những video này sẽ được giấu tin bằng trải phổ. Những bản sao của những video này khi chưa giấu cũng được giữ lại đại diện cho video chưa giấu tin. Tất cả video (28) sẽ đều được tiến hành trích xuất đặc trưng. Cross validation (Kiểm tra chéo), một phương pháp dùng để chọn ra trainning set tốt nhất từ một tập rộng, sẽ được dùng để chọn ra các chuỗi hay vector đặc trưng đại diện cho các lớp khác nhau. Ý tưởng của phương pháp này là chọn ngẫu nhiên n chuỗi video từ trainning set ban đầu và tiến hành dự đoán xác suất false-negative và false-positive cho 14-n video còn lại. N video trước làm trainning data cho 14-n trainning video sau. Quá trình lăp lại 1 số lần cho đến khi xác suât false-negative và false-positive đủ nhỏ.

## ***IV. Kết luận***

Như vậy, chúng ta đã tìm hiểu được cách thuật toán phát hiện giấu tin trong video hoạt động như thế nào. Bảng dưới tóm tắt lại thuật toán này.

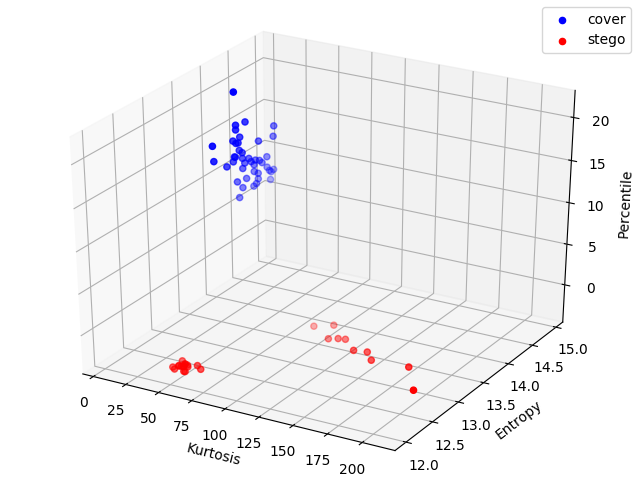
|  |  |  |
| --- | --- | --- |
| Định nghĩa | N | Số lượng frames |
| Xi(,) | Frame thứ i của video |
| Yi(,) | Bản tích hợp của Frame thứ i |
| Ei(,) | Ước lượng watermark của frame thứ i |
| Oi | Kết quả của việc phân loại cho frame thứ i |
| Coll() | Collusion attack trên 2L+1 frames |
| Patt() | Phân loại mẫu |
| Thuật toán | for i={1,2,. . . ,N}  Yi(,) := Coll(Xi(,))  Ei(,) := Xi(,) − Yi(,)  Oi(,) := P att(Ei(,))  end | |

Bảng 1: Bảng tóm tắt phương pháp steganalysis tổng thể kết hợp việc tích hợp tuyến tính và phân loại.

# **C. DEMO**

- Demo phát hiện giấu tin trong video sử dụng Collusion Attack

- Source code demo sẽ được public tại <http://github.com/luffymonster/video-stenalysis>

-

# ***Tài Liệu Tham Khảo***

[0] Histogram-Based Information Extraction, Gamal Seedahmed, University of Khartoum, March 2016

[1] Budia, U., Kundur, D., and Zourntos, T. 2006. Digital Video Steganalysis Exploiting Statistical Visibility in the Temporal Domain. IEEE Transactions on Information Forensics and Security. 1, 4 (Dec. 2006), 502-516.

[2] Cox, I., Kilian, J., Leighton, F., and Shamoon, T. 1997. Secure Spread Spectrum Watermarking for Multi-media. IEEE Transactions on Image Processing. 6, 12 (Dec. 1997), 1673-1687.

[3] Marvel, L. M., Boncelet Jr. C. G., and Retter, C. T. 1999. Spread Spectrum Image Steganography. IEEE Transactions on Image Processing. 8, 8 (Aug. 1999), 1075-1083.

[4] Jainsky, J. S., Kundur, D., and Halverson, D. R. Towards Digital Video Steganalysis using Asymptotic Memoryless Detection. In Proceedings of the 9th International Workshop on Multimedia and Security (Dallas, USA, September 20 - 21, 2007). 161-168.

[5] Lehmann, E. L., and Romano, J. P. 2005. Testing Statistical Hypotheses. 3rd Edition. Springer Texts in Statistics

[6] Su, Y., Zhang, C., Wang, L., and Zhang, C. A New Video Steganalysis based on Mode Detection. In Proceedings of the International Conference on Audio, Language and Image Processing (Shanghai, China, July 7 - 9, 2008). 1507-1510.

[7] Pankajakshan, V., and Ho, A. T. S. Improving Video Steganalysis using Temporal Correlation. In Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (Splendor Kaohsiung, Taiwan, November 26 - 28, 2007). 1, 287-290.

[8] Wang, Y., Osterman, J., and Zhang, Y.-Q. 2001. Video Processing and Communications. Prentice Hall.

[9] Kancherla, K., and Mukkamala, S. Video Steganalysis using Spatial and Temporal Redundancies. In Proceedings of the International Conference on High Performance Computing and Simulation (Leipzig, Germany, June 21 - 24, 2009). 200207.

[10] Zhang, C., Su, Y., and Zhang, C. 2008. Video Steganalysis based on Aliasing Detection. Electronic Letters. 44, 13 (Jun. 2008), 801-803.

[11] Liu, B., Liu, F., and Wang, P. Inter-frame Correlation based Compression Video Steganalysis. In Proceedings of the Congress on Image and Signal Processing (Sanya, China, May 27 - 30, 2008), 3, 42-46.

[12] Wang H, Wang S (2004) Cyber warfare: steganography vs. steganalysis. Commun the ACM 47(10):76–82.

[13] Johnson NF, Jajodia S (1998) Steganalysis: The investigation of hidden information. In: Information Technology Conference 113–116.

[14] Su Y, Zhang C, Wang L, Zhang C (2008) A new video steganalysis based on mode detection. In: International Conference on Audio, Language and Image Processing (ICALIP 2008) 1507–1510.

[15] ZakerN, HamzehA (2012) A novelsteganalysis forTPVD steganographic method based on differences of pixel difference histogram. Multimed Tools Appl 58(1):147–166.

[16]SuY,ZhangC,ZhangC(2011)Avideosteganalyticalgorithmagainstmotion-vector-basedsteganography. Signal Process 91(8):1901–1909.

[17] Xu C, Ping X (2007) A steganographic algorithm in uncompressed video sequence based on difference between adjacent frames. In: Fourth International Conference on Image and Graphics (ICIG) 297–302.

[18] Yang M,Bourbakis N (2005) A highbitrate information hiding algorithm for digital videocontentunder H. 264/AVC compression. In: 48th Midwest Symposium on Circuits and Systems 935–938.

[19] Chae JJ, Manjunath BS (1999) Data hiding in video. In: Proceedings of International Conference on Image Processing (ICIP 99) 311–315.

[20] Hanafy AA, Salama GI, Mohasseb YZ (2008) A secure covert communication model based on video steganography. In: Military Communications Conference (MILCOM 2008) 1–6.

[21] Hu S, KinTak U (2011) A Novel Video Steganography Based on Non-uniform Rectangular Partition. In: IEEE 14th International Conference on Computational Science and Engineering (CSE) 57–61.

[22] Mansouri J, Khademi M (2009) An adaptive scheme for compressed video steganography using temporal and spatial features of the video signal. Int J Imaging Syst Technol 19(4):306–315.

[23] Sherly AP, Amritha PP (2010) A Compressed Video Steganography using TPVD. Int J of Database Manag Syst 2 (3). doi:5121/ijdms.2010.2307 67.

[24] Sherly AP, Amritha PP (2010) A Compressed Video Steganography using TPVD. Int J of Database Manag Syst 2 (3). doi:5121/ijdms.2010.2307 67.

[25] Balaji R, Naveen G (2011) Secure data transmission using video Steganography. In: IEEE International Conference on Electro/Information Technology (EIT) 1–5.

[26] Eltahir ME, Kiah LM, Zaidan BB, Zaidan AA (2009) High rate video streaming steganography. In: International Conference on Future Computer and Communication (ICFCC 2009) 672–675.

[27] Jalab H, Zaidan AA, Zaidan BB (2009)Frame selectedapproach for hidingdata within MPEG video using bit plane complexity segmentation. J Comput 1(1):108–113.

[28] Shou-Dao W, Chuang-Bai X, Yu L A High Bitrate Information Hiding Algorithm for Video in Video.

[29] Yang M,Bourbakis N (2005) A highbitrate information hiding algorithm for digital videocontentunder H. 264/AVC compression. In: 48th Midwest Symposium on Circuits and Systems 935–938.

[30] Cheddad A, Condell J, Curran K, Mc Kevitt P (2009) A skin tone detection algorithm for an adaptive approach to steganography. Signal Process 89(12):2465–2478.

[31] Xu C, Ping X, Zhang T (2006) Steganography in compressed video stream. In: First International Conference on Innovative Computing, Information and Control (ICICIC’06) 269–272.

Table entry for ***z*** is the area under the standard normal curve to the left of ***z***.

Table entry

***z***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***z*** | **.00** | **.01** | **.02** | **.03** | **.04** | **.05** | **.06** | **.07** | **.08** | **.09** |
| –3.4 | .0003 | .0003 | .0003 | .0003 | .0003 | .0003 | .0003 | .0003 | .0003 | .0002 |
| –3.3 | .0005 | .0005 | .0005 | .0004 | .0004 | .0004 | .0004 | .0004 | .0004 | .0003 |
| –3.2 | .0007 | .0007 | .0006 | .0006 | .0006 | .0006 | .0006 | .0005 | .0005 | .0005 |
| –3.1 | .0010 | .0009 | .0009 | .0009 | .0008 | .0008 | .0008 | .0008 | .0007 | .0007 |
| –3.0 | .0013 | .0013 | .0013 | .0012 | .0012 | .0011 | .0011 | .0011 | .0010 | .0010 |
| –2.9 | .0019 | .0018 | .0018 | .0017 | .0016 | .0016 | .0015 | .0015 | .0014 | .0014 |
| –2.8 | .0026 | .0025 | .0024 | .0023 | .0023 | .0022 | .0021 | .0021 | .0020 | .0019 |
| –2.7 | .0035 | .0034 | .0033 | .0032 | .0031 | .0030 | .0029 | .0028 | .0027 | .0026 |
| –2.6 | .0047 | .0045 | .0044 | .0043 | .0041 | .0040 | .0039 | .0038 | .0037 | .0036 |
| –2.5 | .0062 | .0060 | .0059 | .0057 | .0055 | .0054 | .0052 | .0051 | .0049 | .0048 |
| –2.4 | .0082 | .0080 | .0078 | .0075 | .0073 | .0071 | .0069 | .0068 | .0066 | .0064 |
| –2.3 | .0107 | .0104 | .0102 | .0099 | .0096 | .0094 | .0091 | .0089 | .0087 | .0084 |
| –2.2 | .0139 | .0136 | .0132 | .0129 | .0125 | .0122 | .0119 | .0116 | .0113 | .0110 |
| –2.1 | .0179 | .0174 | .0170 | .0166 | .0162 | .0158 | .0154 | .0150 | .0146 | .0143 |
| –2.0 | .0228 | .0222 | .0217 | .0212 | .0207 | .0202 | .0197 | .0192 | .0188 | .0183 |
| –1.9 | .0287 | .0281 | .0274 | .0268 | .0262 | .0256 | .0250 | .0244 | .0239 | .0233 |
| –1.8 | .0359 | .0351 | .0344 | .0336 | .0329 | .0322 | .0314 | .0307 | .0301 | .0294 |
| –1.7 | .0446 | .0436 | .0427 | .0418 | .0409 | .0401 | .0392 | .0384 | .0375 | .0367 |
| –1.6 | .0548 | .0537 | .0526 | .0516 | .0505 | .0495 | .0485 | .0475 | .0465 | .0455 |
| –1.5 | .0668 | .0655 | .0643 | .0630 | .0618 | .0606 | .0594 | .0582 | .0571 | .0559 |
| –1.4 | .0808 | .0793 | .0778 | .0764 | .0749 | .0735 | .0721 | .0708 | .0694 | .0681 |
| –1.3 | .0968 | .0951 | .0934 | .0918 | .0901 | .0885 | .0869 | .0853 | .0838 | .0823 |
| –1.2 | .1151 | .1131 | .1112 | .1093 | .1075 | .1056 | .1038 | .1020 | .1003 | .0985 |
| –1.1 | .1357 | .1335 | .1314 | .1292 | .1271 | .1251 | .1230 | .1210 | .1190 | .1170 |
| –1.0 | .1587 | .1562 | .1539 | .1515 | .1492 | .1469 | .1446 | .1423 | .1401 | .1379 |
| –0.9 | .1841 | .1814 | .1788 | .1762 | .1736 | .1711 | .1685 | .1660 | .1635 | .1611 |
| –0.8 | .2119 | .2090 | .2061 | .2033 | .2005 | .1977 | .1949 | .1922 | .1894 | .1867 |
| –0.7 | .2420 | .2389 | .2358 | .2327 | .2296 | .2266 | .2236 | .2206 | .2177 | .2148 |
| –0.6 | .2743 | .2709 | .2676 | .2643 | .2611 | .2578 | .2546 | .2514 | .2483 | .2451 |
| –0.5 | .3085 | .3050 | .3015 | .2981 | .2946 | .2912 | .2877 | .2843 | .2810 | .2776 |
| –0.4 | .3446 | .3409 | .3372 | .3336 | .3300 | .3264 | .3228 | .3192 | .3156 | .3121 |
| –0.3 | .3821 | .3783 | .3745 | .3707 | .3669 | .3632 | .3594 | .3557 | .3520 | .3483 |
| –0.2 | .4207 | .4168 | .4129 | .4090 | .4052 | .4013 | .3974 | .3936 | .3897 | .3859 |
| –0.1 | .4602 | .4562 | .4522 | .4483 | .4443 | .4404 | .4364 | .4325 | .4286 | .4247 |

Table entry for ***z*** is the area under the standard normal curve to the left of ***z***.

Table entry

***z***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***z*** | **.00** | **.01** | **.02** | **.03** | **.04** | **.05** | **.06** | **.07** | **.08** | **.09** |
| 0.0 | .5000 | .5040 | .5080 | .5120 | .5160 | .5199 | .5239 | .5279 | .5319 | .5359 |
| 0.1 | .5398 | .5438 | .5478 | .5517 | .5557 | .5596 | .5636 | .5675 | .5714 | .5753 |
| 0.2 | .5793 | .5832 | .5871 | .5910 | .5948 | .5987 | .6026 | .6064 | .6103 | .6141 |
| 0.3 | .6179 | .6217 | .6255 | .6293 | .6331 | .6368 | .6406 | .6443 | .6480 | .6517 |
| 0.4 | .6554 | .6591 | .6628 | .6664 | .6700 | .6736 | .6772 | .6808 | .6844 | .6879 |
| 0.5 | .6915 | .6950 | .6985 | .7019 | .7054 | .7088 | .7123 | .7157 | .7190 | .7224 |
| 0.6 | .7257 | .7291 | .7324 | .7357 | .7389 | .7422 | .7454 | .7486 | .7517 | .7549 |
| 0.7 | .7580 | .7611 | .7642 | .7673 | .7704 | .7734 | .7764 | .7794 | .7823 | .7852 |
| 0.8 | .7881 | .7910 | .7939 | .7967 | .7995 | .8023 | .8051 | .8078 | .8106 | .8133 |
| 0.9 | .8159 | .8186 | .8212 | .8238 | .8264 | .8289 | .8315 | .8340 | .8365 | .8389 |
| 1.0 | .8413 | .8438 | .8461 | .8485 | .8508 | .8531 | .8554 | .8577 | .8599 | .8621 |
| 1.1 | .8643 | .8665 | .8686 | .8708 | .8729 | .8749 | .8770 | .8790 | .8810 | .8830 |
| 1.2 | .8849 | .8869 | .8888 | .8907 | .8925 | .8944 | .8962 | .8980 | .8997 | .9015 |
| 1.3 | .9032 | .9049 | .9066 | .9082 | .9099 | .9115 | .9131 | .9147 | .9162 | .9177 |
| 1.4 | .9192 | .9207 | .9222 | .9236 | .9251 | .9265 | .9279 | .9292 | .9306 | .9319 |
| 1.5 | .9332 | .9345 | .9357 | .9370 | .9382 | .9394 | .9406 | .9418 | .9429 | .9441 |
| 1.6 | .9452 | .9463 | .9474 | .9484 | .9495 | .9505 | .9515 | .9525 | .9535 | .9545 |
| 1.7 | .9554 | .9564 | .9573 | .9582 | .9591 | .9599 | .9608 | .9616 | .9625 | .9633 |
| 1.8 | .9641 | .9649 | .9656 | .9664 | .9671 | .9678 | .9686 | .9693 | .9699 | .9706 |
| 1.9 | .9713 | .9719 | .9726 | .9732 | .9738 | .9744 | .9750 | .9756 | .9761 | .9767 |
| 2.0 | .9772 | .9778 | .9783 | .9788 | .9793 | .9798 | .9803 | .9808 | .9812 | .9817 |
| 2.1 | .9821 | .9826 | .9830 | .9834 | .9838 | .9842 | .9846 | .9850 | .9854 | .9857 |
| 2.2 | .9861 | .9864 | .9868 | .9871 | .9875 | .9878 | .9881 | .9884 | .9887 | .9890 |
| 2.3 | .9893 | .9896 | .9898 | .9901 | .9904 | .9906 | .9909 | .9911 | .9913 | .9916 |
| 2.4 | .9918 | .9920 | .9922 | .9925 | .9927 | .9929 | .9931 | .9932 | .9934 | .9936 |
| 2.5 | .9938 | .9940 | .9941 | .9943 | .9945 | .9946 | .9948 | .9949 | .9951 | .9952 |
| 2.6 | .9953 | .9955 | .9956 | .9957 | .9959 | .9960 | .9961 | .9962 | .9963 | .9964 |
| 2.7 | .9965 | .9966 | .9967 | .9968 | .9969 | .9970 | .9971 | .9972 | .9973 | .9974 |
| 2.8 | .9974 | .9975 | .9976 | .9977 | .9977 | .9978 | .9979 | .9979 | .9980 | .9981 |
| 2.9 | .9981 | .9982 | .9982 | .9983 | .9984 | .9984 | .9985 | .9985 | .9986 | .9986 |
| 3.0 | .9987 | .9987 | .9987 | .9988 | .9988 | .9989 | .9989 | .9989 | .9990 | .9990 |
| 3.1 | .9990 | .9991 | .9991 | .9991 | .9992 | .9992 | .9992 | .9992 | .9993 | .9993 |
| 3.2 | .9993 | .9993 | .9994 | .9994 | .9994 | .9994 | .9994 | .9995 | .9995 | .9995 |
| 3.3 | .9995 | .9995 | .9995 | .9996 | .9996 | .9996 | .9996 | .9996 | .9996 | .9997 |