HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đồng Thanh Tú

TÌM HIỂU HỆ THỐNG DATA WAREHOUSE VÀ ỨNG DỤNG CỦA CHÚNG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2021

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đồng Thanh Tú

TÌM HIỂU HỆ THỐNG DATA WAREHOUSE VÀ ỨNG DỤNG CỦA CHÚNG

Chuyên ngành: Hệ thống Thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. HỒ VĂN CANH

HÀ NỘI - 2021

LỜI CẢM ƠN

Để hoàn thành chương trình cao học và viết luận văn này học viên đã nhận được sự hướng dẫn, giúp đỡ và góp ý nhiệt tình của các thầy cô Khoa sau đại học, Học viện Công nghệ Bưu chính viễn thông.

Trước hết, học viên xin chân thành cảm ơn các thầy, cô giáo, cán bộ thuộc Khoa sau đại học, Học viện Công nghệ Bưu chính viễn thông, đặc biệt là những thầy cô đã tận tình dạy bảo cho học viên trong suốt thời gian học tại trường.

Học viên xin chân thành cảm ơn Thầy giáo **TS Hồ Văn Canh**, Cục Kỹ thuật nghiệp vụ, Bộ Công an đã dành nhiều thời gian và tâm huyết hướng dẫn khoa học cho học viên trong quá trình thực hiện luận văn này.

Nhân đây, học viên xin chân thành cảm ơn các anh, các chị và các bạn lớp Cao học M19CQIS01-B, chuyên ngành Hệ thống thông tin, Học viện Cộng nghệ Bưu chính viễn thông đã luôn động viên, giúp đỡ và nhiệt tình chia sẻ với học viên những kinh nghiệm học tập, công tác trong suốt khoá học.

Cuối cùng, học viên muốn gửi lời cảm ơn tới gia đình, bạn bè, những người thân luôn bên cạnh và động viên học viên trong suốt quá trình thực hiện luận văn tốt nghiệp.

Mặc dù học viên đã cố gắng hoàn thiện luận văn bằng tất cả sự nhiệt tình và năng lực của mình, song luận văn này không thể tránh khỏi những thiếu sót, kính mong được sự chỉ dẫn của các quý thầy cô và các bạn.

Hà Nội, ngày 10 tháng 9 năm 2021
Học viên

Đồng Thanh Tú

LỜI CAM ĐOAN

Luận văn Thạc sĩ "**TÌM HIỂU HỆ THỐNG DATA WAREHOUSE VÀ ỨNG DỤNG CỦA CHÚNG** chuyên ngành Hệ thống thông tin là công trình của riêng học viên, chưa được sử dụng để bảo vệ một học vị nào.

Học viên xin cam đoan rằng số liệu và kết quả nghiên cứu trong luận văn này là trung thực và không trùng lặp với các đề tài khác. Luận văn đã sử dụng thông tin từ nhiều nguồn dữ liệu khác nhau và các thông tin trích dẫn trong luận văn đã được chỉ rõ nguồn gốc.

Hà Nội, ngày 10 tháng 9 năm 2021
Tác giả

Đồng Thanh Tú

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ BẢNG BIỂU

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial Intelligent	Trí tuệ nhân tạo
BD	Big Data	Dữ liệu lớn
BI	Business Intelligence	Quản trị thông minh
BMP	Bitmap	Định dạng ảnh Bitmap
CSDL		Cơ sở dữ liệu
DBMS	Distributed Database Management System	Hệ quản trị cơ sở dữ liệu phân tán
DDB	Distributed DataBase	Cơ sở dữ liệu phân tán
DDBS	Distributed DataBase System	Hệ cơ sở dữ liệu phân tán
DH	Data Hiding	Kỹ thuật giấu tin
DICOM	Digital Image and Communication in Medicine	Tập hợp các chuẩn dùng trong xử lý truyền tải thông tin
DT	Distortion techniques	Kỹ thuật làm méo vật mang
DWH	Data warehouses	Kho dữ liệu
DWHT	Data warehouse Technology	Công nghệ kho dữ liệu
ETL	Extraction Transformation Loading	Chuyển đổi dữ liệu được trích chọn
FD	Frequeney Domain	Miền tần số
GCD	Greatest Common Divisor	Ước số chung lớn nhất
IoT	Internet of Things	Internet của vạn vật

JPEG	Joint Photographic Experts Group	Định dạng của nhóm chuyên gia nhiếp ảnh
LSB	Leastest Signifieant Bit	Bít có ý nghĩa thấp nhất
MD	Meta Data	Siêu dữ liệu
MSE	Mean Square Error	Sai số bình phương trung bình
OLAP	OnLine Analytical Processing	Xử lý phân tích trực tuyến
PSNR	Peak Signal of Noise Ratio	Tỉ số tín hiệu cực đại trên nhiễu
SD	Spacial Domain	Miền không gian
SST	Spread Spectram Techniques	Các kỹ thuật trải phổ

MỞ ĐẦU

Trong thời đại cách mạng công nghiệp 4.0, khi mà những yếu tố cốt lõi là Trí tuệ nhân tạo (AI), vạn vật kết nối – Internet of Things (IoT) và dữ liệu lớn (Big Data) luôn là xu hướng hàng đầu giúp tự động hóa xử trí và trao đổi thông tin, dữ liệu. Big Data chứa trong mình rất nhiều thông tin quý giá mà nếu trích xuất thành công, nó sẽ giúp rất nhiều cho việc: kinh doanh, nghiên cứu khoa học, dự đoán các dịch bệnh sắp phát sinh... Chính vì thế, những dữ liệu này phải được thu thập, tổ chức, lưu trữ, tìm kiếm, chia sẻ theo một cách khác so với bình thường.

Các doanh nghiệp để có được sự thành công của mình, nhất thiết cần vào sự quyết định đúng đắn của các nhà quản trị doanh nghiệp, việc quyết định dựa trên những số liệu liên quan và cách tốt nhất để có nhanh các dữ liệu liên quan, tìm kiếm các minh chứng này là phải có một kho dữ liệu và đó là lý do sự ra đời của Data warehouse.

Data warehouse thực hiện quá trình truy cập dữ liệu từ các nguồn không đồng nhất, làm sạch, lọc và chuyển đổi dữ liệu, lưu trữ dữ liệu theo cấu trúc để dễ dàng truy cập, hiểu rõ và sử dụng. Từ nguồn dữ liệu ở khắp mọi nơi, hệ thống sẽ kiểm soát và ra quyết định cụ thể theo yêu cầu.

Kho dữ liệu là một hướng công nghệ mới được sử dụng phổ biến cho các bài toán lớn hiện nay như: y tế, bảo hiểm, ngân hàng, dân số, viễn thông.... Việc xây dựng kho dữ liệu không những giúp lưu trữ một lượng thông tin lớn hàng ngày mà còn giúp cho các nhà quản lý có thể trích rút nguồn tài nguyên một cách nhanh chóng, chính xác. Đây cũng là kiến thức rất hữu ích và cần thiết để có thể khai thác ngày một hiệu quả các thành tựu tin học.

Với mục đích, đưa những tiến bộ khoa học, công nghệ vào phục vụ cho cuộc sống, học viên xin chọn đề tài nghiên cứu "*Tìm hiểu hệ thống Data warehouse và ứng dụng của chúng*".

Luận văn tập trung vào nghiên cứu tổng quan về Data warehouse và xây dựng một phần mềm ứng dụng nhằm tìm kiếm các thông tin liên quan đến Bảo hiểm

y tế ngành Công an được lưu trong cơ sở dữ liệu của Hệ thống thông tin giám định bảo hiểm y tế.

Nội dung của luận văn bao gồm 03 chương:

Chương 1: TỔNG QUAN VỀ DATA WAREHOUSE

Chương 2: XÂY DỰNG THUẬT TOÁN GIẤU THÔNG TIN MẬT TRONG CƠ SỞ DỮ LIỆU DATA WAREHOUSE

Chương 3: ĐỀ XUẤT THUẬT TOÁN GIẤU THÔNG TIN MẬT VÀ ỨNG DỤNG TRONG NGÀNH Y TẾ

Cuối cùng là phần kết luận và các tài liệu tham khảo.

Chương 1 - TỔNG QUAN VỀ DATA WAREHOUSE

1.1. Khái niệm CSDL phân tán, kho dữ liệu

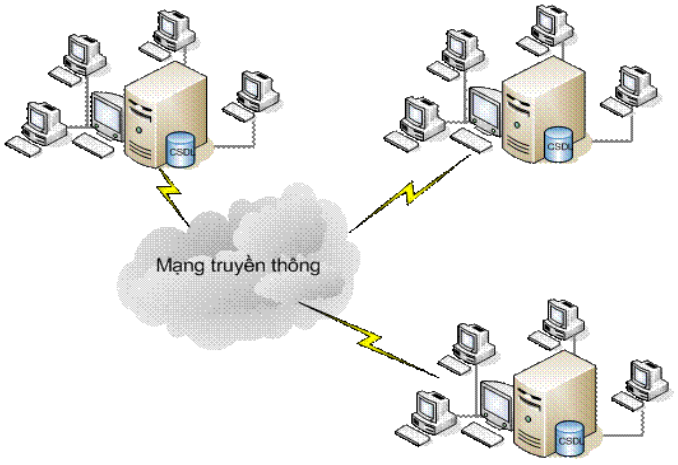
1.1.1. Mô hình CSDL phân tán

Cơ sở dữ liệu là một trong những nội dung rất được quan tâm trong quá trình xây dựng các hệ thống thông tin, đặc biệt là hệ thông thông tin quản lý.

Cơ sở dữ liệu phân tán (*Distributed DataBase – DDB*) là một tập hợp nhiều cơ sở dữ liệu có liên đới logic và được phân bố trên một mạng máy tính. Với khái niệm này, có 02 thuật ngữ quan trọng trong các định nghĩa là "liên đới logic" và "phân bố trên một mạng máy tính".

- Liên đới logic: Toàn bộ dữ liệu của cơ sở dữ liệu phân tán có một số các thuộc tính ràng buộc chúng với nhau. Điều này giúp ta có thể phân biệt một cơ sở dữ liệu phân tán với một tập hợp cơ sở dữ liệu cục bộ hoặc các tập tin lưu trữ tại các vị trí khác nhau trong một mạng máy tính.

- Phân bố trên một mạng máy tính: Toàn bộ dữ liệu của cơ sở dữ liệu phân tán không được lưu trữ ở một nơi mà lưu trữ trên nhiều trạm thuộc mạng máy tính. Điều này giúp phân biệt cơ sở dữ liệu phân tán với cơ sở dữ liệu tập trung đơn lẻ.



Hình 1.1. Sơ đồ cơ sở dữ liệu phân tán

Hệ quản trị cơ sở dữ liệu phân tán (*Distributed Database Management System D – DBMS*) được định nghĩa là một hệ thống phần mềm cho phép quản lý các hệ cơ sở dữ liệu phân tán và làm cho sự phân tán trở nên "trong suốt" đối với người sử dụng.

Hệ cơ sở dữ liệu phân tán (*Distributed DataBase System – DDBS*) được xây dựng dựa trên hai công nghệ cơ bản là cơ sở dữ liệu và mạng máy tính. Một hệ cơ sở dữ liệu phân tán không phải là một "tập hợp các tập tin" được lữu trữ riêng rẽ tại mỗi nút của một mạng máy tính. Để tạo ra một hệ cơ sở dữ liệu phân tán các tập tin không chỉ có liên đới logic mà chúng còn phải có cấu trúc và được truy xuất qua một giao diện chung. Có 02 hệ cơ sở dữ liệu phân tán:

- Hệ cơ sở dữ liệu phân tán không thuần nhất: cơ sở dữ liệu cục bộ ở các nơi (Site) không dùng chung một hệ quản trị cơ sở dữ liệu.

- Hệ cơ sở dữ liệu phân tán thuần nhất: cơ sở dữ liệu cục bộ ở tất cả các nơi (Site) đều dùng chung một hệ quản trị cơ sở dữ liệu.

1.1.2. Định nghĩa kho dữ liệu (Data warehouses)

Nếu như cơ sở dữ liệu được ví như cái tủ sách cá nhân, nơi người ta thường xuyên tra cứu, cập nhật, hiệu đính, ghi chú vào lề, thêm mới hoặc chuyển sách đi, thì kho dữ liệu lại được so sánh với thư viện quốc gia, nơi các tài liệu kinh điển được đưa đến liên tục để lưu trữ và tham khảo, không ai sửa chữa hoặc chuyển chúng qua chỗ nào khác cả.

Kho dữ liệu (Data warehouse) là tuyển tập các cơ sở dữ liệu tích hợp, hướng chủ đề, được thiết kế để hỗ trợ cho chức năng trợ giúp quyết định. Kho dữ liệu thường rất lớn tới hàng trăm Girabyte hay thậm chí hàng Terabyte. Kho dữ liệu được xây dựng để tiện lợi cho việc truy cập theo nhiều nguồn, nhiều kiểu dữ liệu khác nhau sao cho có thể kết hợp được cả những ứng dụng của các công nghệ hiện đại và kế thừa được từ những hệ thống đã có sẵn từ trước.

Kho dữ liệu được hiểu là kho lưu trữ dữ liệu bằng thiết bị điện tử của một tổ chức. Kho dữ liệu chính là hệ quản trị cơ sở dữ liệu chuyên dùng để tạo báo cáo và

phân tích dữ liệu. Nó vừa hỗ trợ các truy vấn phức tạp, vừa là điểm tập trung dữ liệu từ nhiều nguồn khác nhau để có được thông tin phân tích đầy đủ nhất.

Cấu trúc của một kho dữ liệu bao gồm ba tầng:

- Tầng đáy: là nơi cung cấp dịch vụ và lấy dữ liệu từ nhiều nguồn khác nhau, sau đó chuẩn hóa, làm sạch và lưu trữ dữ liệu đã tập trung.
- Tầng giữa: cung cấp các dịch vụ để thực hiện các thao tác với kho dữ liệu gọi là dịch vụ OLAP (OLAP server). Có thể cài đặt bằng Relational OLAP, Multidimensional OLAP hay kết hợp cả hai mô hình trên Hybrid OLAP.
- Tầng trên cùng: nơi chứa các câu truy vấn, báo cáo, phân tích.

Hình 1.2. Sơ đồ chung kho dữ liệu

Hình 1.2 mô tả kiến trúc cơ bản của Data warehouse, dựa trên nguyên tắc là xây dựng một kho dữ liệu thống nhất từ nhiều nguồn dữ liệu khác nhau để phục vụ truy vấn.

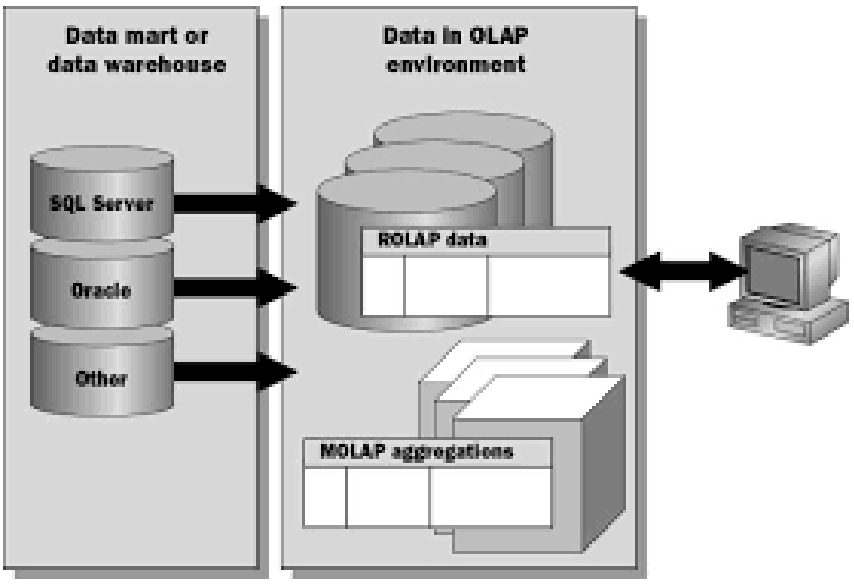
Như vậy có thể thấy, Data warehouse là một cơ sở dữ liệu quan hệ được thiết kế hướng tới truy vấn, phân tích dữ liệu một cách chính xác trên tập dữ liệu lớn. Data warehouse còn là nơi lưu trữ các dữ liệu giao dịch có tính lịch sử được thu thập, xử lý từ nhiều nguồn khác nhau. Data warehouse ngoài một cơ sở dữ liệu, thì cần rất nhiều thành phần bổ sung để tạo nên một cấu trúc hoàn chỉnh, gồm tập hợp các bộ công cụ có nhiệm vụ xử lý, thu thập và đưa dữ liệu vào kho, công cụ tạo báo

cáo, data mining… Data warehouse là yếu tố cơ bản, đóng vai trò then chốt trong việc tập hợp, xử lý dữ liệu thô, do dữ liệu có thể đến từ nghiều nguồn khác nhau, dẫn đến việc dữ liệu không đồng nhất. Bên cạnh đó, khối lượng dữ liệu sẽ tăng nhanh theo thời gian, điều này dẫn đến quá tải và giảm đáng kể khả năng truy xuất. Việc xây dựng Data warehouse là công đoạn đầu tiên và cơ bản trong quá trình tích hợp hệ thống báo cáo quản trị thông minh, từ đó hệ thống báo cáo có thể truy xuất, xử lý dữ liệu một cách nhanh chóng và dễ dàng.

Công nghệ kho dữ liệu (*Data warehouse Technology*) là tập các phương pháp, kỹ thuật và các công cụ có thể kết hợp, hỗ trợ nhau để cung cấp thông tin cho người sử dụng trên cơ sở tích hợp từ nhiều nguồn dữ liệu, nhiều môi trường khác nhau.

1.2. Xử lý phân tích trực tuyến

Kho dữ liệu cho phép người dùng ở mức quản lý, ra quyết định thực hiện các phép phân tích tương tác với data bằng hệ thống xử lý phân tích trực tuyến *(OnLine Analytical Processing - OLAP).* Trong khoa học máy tính, xử lý phân tích trực tuyến là một phương pháp để xử lý các truy vấn về phân tích khối lượng dữ liệu lớn, nhiều chiều mà nếu cho thực thi các truy vấn này trong hệ thống cơ sở dữ liệu thông thường sẽ không thể cho kết quả hoặc sẽ mất rất nhiều thời gian.



Hình 1.3. Sơ đồ xử lý phân tích trực tuyến

Căn cứ vào cách thức lưu trữ dữ liệu, người ta thường tiếp cận mô hình dữ liệu đa chiều theo 3 hướng sau:

- OLAP kiểu quan hệ (*Relational OLAP – ROLAP)* lưu trữ dữ liệu trong cơ sở dữ liệu quan hệ, dùng câu lệnh SQL để thực hiện các tính năng của OLAP.

- OLAP đa chiều (*Multicắt lớp OLAP – MOLAP*) lưu trữ dữ liệu dưới dạng file có cấu trúc đặc thù và thực hiện các tính năng OLAP trên cấu trúc này. Mặc dù bị hạn chế về lượng dữ liệu lưu trữ và xử lý được so với ROLAP, MOLAP thường cho hiệu năng tốt hơn trong các phép truy vấn hoặc tổng hợp số liệu vì dữ liệu được thiết kế tối ưu cho truy vấn OLAP trong khi ROLAP phải thông qua cơ sở dữ liệu.

- OLAP lai (*Hybrid OLAP – HOLAP*) kết hợp hai công nghệ ROLAP và MOLAP nói trên, tận dụng khả năng lưu trữ của OLAP và khả năng xử lý của MOLAP. Ví dụ HOLAP sẽ lưu dữ liệu chi tiết trên cơ sở dữ liệu quan hệ còn dữ liệu tổng hợp hơn để truy vấn cho người dùng được lưu trên không gian MOLAP.

1.2.1. Các tính năng của OLAP

Như đã nói ở trên, tính chất cơ bản của mô hình dữ liệu đa chiều là cho phép người dùng quan sát dữ liệu trên nhiều phương diện khác nhau, ở các mức độ chi tiết khác nhau. OLAP cung cấp một số tính năng cho phép thực hiện điều đó, cụ thể:

- Tính năng nhìn xa (*roll-up*) biến tiêu chí từ mức chi tiết sang mức tổng hợp để hiển thị cho người dùng, được thực hiện khi đi từ mức thấp lên mức cao trong cây phân cấp hoặc giảm số cắt lớp xuống.

- Tính năng đào sâu (*drill-down*) thực hiện ngược lại với nhìn xa, tức là đi từ mức tổng hợp cao đến mức chi tiết hơn.

- Tính năng đảo chiều (*pivot hoặc rotate*) biến hàng thành cột, cột thành hàng giúp cung cấp cho người dùng một cách thể hiện dữ liệu khác.

- Tính năng cắt lát mỏng (*slice*) thực hiện cắt lấy dữ liệu một lớp cắt cụ thể trong một cắt lớp.

Ngoài 5 tính năng cơ bản trên, các bộ công cụ OLAP trên thị trường cũng cung cấp thêm một loạt các tính năng hỗ trợ khác như các phép toán số học, thống kê, các phép toán kinh tế…

1.2.2. Các thành phần OLAP sử dụng

Những thành phần mà OLAP sử dụng để thực hiện các dịch vụ bao gồm:

- *Nguồn dữ liệu*: các cơ sở dữ liệu OLAP và các nguồn dữ liệu hợp lệ khác chứa các dữ liệu có thể chuyển đổi thành dữ liệu OLAP trong kho lưu trữ.

- *Kho trung gian*: nơi lưu trữ và xử lý dữ liệu được tập hợp sau đó được sắp xếp, sàng lọc và chuyển đổi thành dữ liệu OLAP hữu ích.

- *Máy chủ lưu trữ*: các máy tính chạy cơ sở dữ liệu liên kết chứa các dữ liệu cho kho lưu trữ và các máy chủ quản lý dữ liệu OLAP.

- *Ứng dụng thông minh*: các bộ công cụ và ứng dụng thực hiện truy vấn dữ liệu OLAP cung cấp các báo cáo, thông tin cho những người ra quyết định.

- *Siêu dữ liệu*: các đối tượng như các bảng, biểu trong cơ sở dữ liệu OLAP, các khối trong kho lưu trữ dữ liệu và các bản ghi mà ứng dụng tham chiếu tới các đoạn dữ liệu khác nhau.

1.3. Dữ liệu Data warehouse

Data warehouse là tập hợp dữ liệu tương đối ổn định, không hay thay đổi, cập nhật theo thời gian, được tích hợp theo hướng chủ đề nhằm hỗ trợ quá trình đưa ra quyết định về mặt quản lý. Một Data warehouse điển hình sẽ:

- Chứa một số lượng lớn dữ liệu có liên quan tới các giao dịch trong quá khứ.

- Được tối ưu hóa cho các thao tác đọc trong các yêu cầu truy vấn dữ liệu, điều này đối lập với các cơ sở dữ liệu trong các hệ thống xử lý tác vụ được thiết kế để hỗ trợ các thao tác thêm, xóa, sửa dữ liệu.

- Được nạp các dữ liệu mới hoặc dữ liệu được cập nhật một cách định kỳ.

- Là nguồn dữ liệu cơ bản cho các ứng dụng quản trị thông minh (*Business Intelligence - BI*).

1.3.1. Các đặc trưng của kho dữ liệu

Kho dữ liệu là một tập hợp dữ liệu hướng chủ đề, toàn vẹn, không bị rò rỉ mất mát và có giá trị lịch sử. Cụ thể các tính chất đó như sau:

- Tính hướng chủ đề (*Subject – oriented*) nghĩa là kho dữ liệu tập trung vào việc phân tích các yêu cầu quản lý ở nhiều cấp độ khác nhau trong quy trình ra

quyết định. Các yêu cầu phân tích này thường rất cụ thể. Ví dụ: các công ty phân phối sẽ quan tâm đến tình hình kinh doanh, doanh nghiệp viễn thông quan tâm đến lưu lượng dịch vụ…

- Tính toàn vẹn (*Integrated*) giải quyết các khó khăn trong việc kết hợp dữ liệu từ nhiều nguồn dữ liệu khác nhau, giải quyết các sai khác về tên trường dữ liệu (dữ liệu khác nhau nhưng tên giống nhau), ý nghĩa dữ liệu (tên giống nhau nhưng dữ liệu khác nhau), định dạng dữ liệu (tên và ý nghĩa giống nhau nhưng kiểu dữ liệu khác nhau).

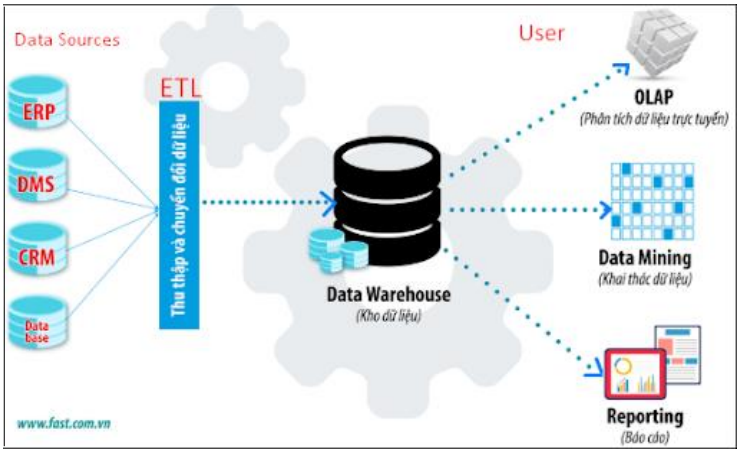
- Tính bất biến (*Nonvolatile*) quy định rằng dữ liệu phải thống nhất theo thời gian (bằng cách hạn chế tối đa sửa đổi hoặc xoá dữ liệu), từ đó làm tăng quy mô dữ liệu lên đáng kể (5-10 năm so với 2 đến 6 tháng như cơ sở dữ liệu thông thường)

- Giá trị lịch sử (*Time – varying*) là khả năng lấy các giá trị khác nhau của cùng một thông tin và thời điểm xảy ra thay đổi. Ví dụ: thông tin địa chỉ, email, số điện thoại của khách hàng có thể thay đổi, nhưng việc thay đổi đó không được phép tác động đến giá trị báo cáo, phân tích thực hiện trước khi sự thay đổi xảy ra.

1.3.2. Kiến trúc hệ thống Data warehouse

Data warehouse là một cơ sở dữ liệu quan hệ được xây dựng cho mục đích truy vấn và phân tích dữ liệu mang tính lịch sử, nó không phải là loại cơ sở dữ liệu giao dịch.

Khác với cơ sở dữ liệu giao dịch thông thường, Data warehouse được bổ sung thêm bộ công cụ kết xuất, chuyển đổi và tích hợp dữ liệu (*Extraction Transformation Loading – ETL*) bộ phân tích dữ liệu trực tuyến OLAP và các công cụ quản trị các tiến trình thu thập dữ liệu. Đặc biệt, Data warehouse được tổ chức nâng cao theo các chủ đề Data Mart



Hình 1.4. Thành phần cơ bản của Dataware house

➢ Tầng xử lý dữ liệu (*Extraction Transformation Loading – ETL*) là tầng thấp nhất, ẩn đi với người dùng cuối, bao gồm 3 bước:

- Bước thu thập (*extract*) gom góp dữ liệu từ nhiều nguồn khác nhau về. Các nguồn này có thể là cơ sở dữ liệu hệ thống nghiệp vụ (MS SQL, mySQL, Oracle, DB2…), cũng có thể là file ở các định dạng khác nhau (CSV, fix-length, excel, XML…), có thể là dữ liệu nội bộ hoặc từ bên ngoài. Một hệ thống xử lý dữ liệu tốt phải đảm bảo tương thích với các nguồn dữ liệu thông dụng này.

- Bước chuẩn hoá (*transform*) biến đổi dữ liệu từ định dạng nguồn sang định dạng của kho dữ liệu, bao gồm các bước nhỏ:

  + Bước dọn dẹp (*cleaning*) xoá các bản ghi bị sai, lỗi và chuyển hoá dữ liệu về định dạng chuẩn chung.
  + Bước tập hợp (*integration*) cắt gọt dữ liệu có chung ý nghĩa từ nhiều nguồn khác nhau về một khung duy nhất.
  + Bước tổng hợp (*aggregation*) tổng hợp dữ liệu dựa vào độ chi tiết của kho dữ liệu.

- Bước nạp dữ liệu (*load*) ghi dữ liệu đã được chuẩn hoá vào kho dữ liệu. Bước này bao gồm cả quá trình cập nhật thay đổi từ hệ thống nghiệp vụ vào kho dữ liệu, đảm bảo số liệu báo cáo luôn được cập nhật.

➢ Tầng kho dữ liệu (*Data warehouse*) đứng ở trung tâm một hệ thống kho dữ liệu làm nhiệm vụ lưu trữ dữ liệu bao quanh tất cả các hoạt động. Kho dữ liệu thường bao gồm một hoặc nhiều phân khu dữ liệu, với mỗi phân khu dữ liệu chính là kho dữ liệu thu nhỏ tập trung vào một nghiệp vụ nhất định nào đó của doanh nghiệp. Ngoài nhiệm vụ lưu trữ dữ liệu, tầng kho dữ liệu còn có một thành phần khác rất quan trọng gọi là siêu dữ liệu (*metadata*). Siêu dữ liệu lại được chia làm 2 nhóm là nhóm siêu dữ liệu nghiệp vụ và siêu dữ liệu kỹ thuật.

- Siêu dữ liệu nghiệp vụ (*business metadata*) mô tả ý nghĩa dữ liệu, các luật và ràng buộc tác động lên dữ liệu.

- Siêu dữ liệu kỹ thuật (*technical metadata*) mô tả cách thức tổ chức, lưu trữ và điều khiển dữ liệu trong hệ thống máy tính.

Trong phạm vi kho dữ liệu, siêu dữ liệu kỹ thuật được sử dụng để mô tả thông tin về kho dữ liệu, về dữ liệu nguồn và các tiến trình xử lý dữ liệu. Cụ thể:

+ Siêu dữ liệu mô tả cấu trúc kho dữ liệu và các phân khu dữ liệu ở mức logic và mức vật lý. Ngoài ra nó còn chứa thông tin bảo mật dữ liệu và các thông tin giám sát.
+ Siêu dữ liệu mô tả dữ liệu nguồn, cũng ở mức logic và vật lý
+ Siêu dữ liệu mô tả các tiến trình xử lý dữ liệu, bao gồm cả gốc gác dữ liệu, các luật thu thập, làm sạch, chuyển hoá dữ liệu.

➢ Tầng khai thác dữ liệu (*User*) chứa các công cụ cho người dùng cuối khai thác, sử dụng các dữ liệu trong kho dữ liệu. Một số công cụ chính:

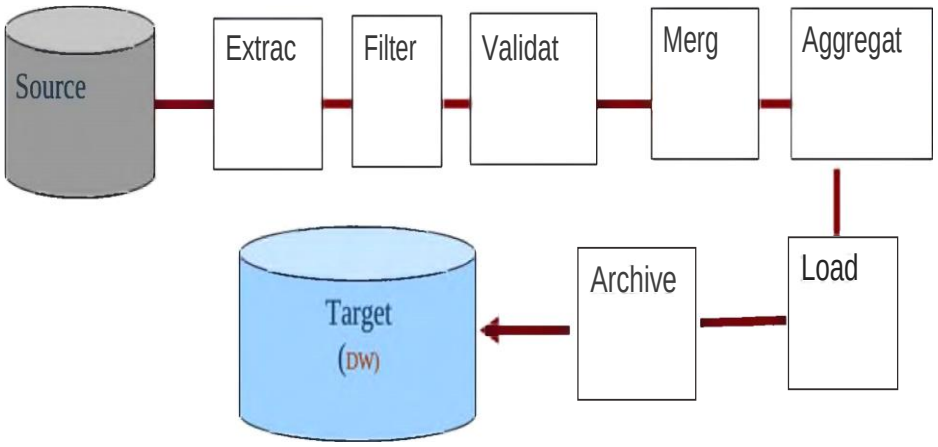
- Báo cáo OLAP (*OLAP tool*) là báo cáo động cho phép người dùng sử dụng các tính năng của OLAP để tạo báo cáo. Các truy vấn đột xuất này được gọi là truy vấn tuỳ biến (*adhoc query*) vì hệ thống không hề được chuẩn bị trước cho thao tác của người dùng. Báo cáo OLAP được sử dụng khi người dùng muốn các thông tin cắt lớp, chuyên sâu hoặc toàn cảnh trước khi ra quyết định.

- Báo cáo tĩnh (*reporting tool*) là các báo cáo có cấu trúc, format, sử dụng truy vấn được định nghĩa trước đó, đôi khi bao gồm cả biểu đồ. Báo cáo tĩnh được sử dụng khi người dùng muốn xem các thông tin đánh giá, điều hành.

- Bộ công cụ khai phá dữ liệu (*data mining*) cho phép người dùng phân tích dữ liệu để tìm ra các thông tin quý giá còn bị ẩn dấu, ví dụ như các xu hướng, các mẫu chung.

Data warehouse và hệ thống phân tích dữ liệu trực tuyến cung cấp các giải pháp để giải quyết các vấn đề trên. Đồng thời, Data warehouse cũng cung cấp phương pháp tiếp cận lưu trữ số liệu từ các nguồn khác nhau vào một kho lưu trữ duy nhất.

1.3.3. Quy trình xây dựng kho dữ liệu



Hình 1.5. Các bước tạo lập kho dữ liệu

Hầu hết mọi hệ thống/dự án công nghệ thông tin, chúng ta đều xem xét dựa trên 3 khía cạnh: cấu trúc (*Structure*), dữ liệu (*Data*), tiến trình (*Process*). Vì vậy, khi xây dựng một hệ thống Data warehouse chúng ta cũng phải cân nhắc 3 khía cạnh này.

- Dữ liệu-*Data*:
  + Cần thông tin gì để hỗ trợ ra quyết định? Ở cấp độ nào?
  + Dữ liệu được lấy từ đâu? Định dạng như thế nào?
  + Độ lớn dữ liệu? Mức độ tăng tưởng dữ liệu như thế nào? Cần bao nhiêu không gian để chứa?

- Cấu trúc-*Structure*:
  + Dữ liệu cần xây dựng theo những chiều nào để phục vụ quá trình phân tích?
  + Cấu trúc dữ liệu nào phù hợp với nhu cầu?
- Tiến trình-*Process*:
  + Tiến trình Extract-Load-Transform được thực hiện như thế nào? Đặt ở đâu? Lập lịch như thế nào?
  + Tiến trình Aggreation cần điều kiện gì để chạy? Entry-point?
  + Có những loại ngoại lệ nào cần xử lý? Ở cấp độ nào?

Tóm lại, Data warehouse là một kho dữ liệu được lưu trữ bằng hệ thống điện tử của một tổ chức, nó được thiết kế để hỗ trợ việc phân tích, tìm kiếm và lập báo cáo. Data warehouse cung cấp các lợi ích sau tới những người dùng:

- Dữ liệu được tổ chức để tạo thuận lợi cho các truy vấn phân tích chứ không phải cho việc xử lý các giao dịch.

- Sự khác biệt về cấu trúc dữ liệu được lưu trữ trên nhiều nguồn dữ liệu không đồng nhất sẽ được giải quyết.

- Những quy tắc thống nhất sẽ được áp dụng khi hợp nhất dữ liệu từ các hệ thống không đồng nhất sang Data warehouse.

- Tính bảo mật và hiệu suất có thể được cải thiện mà không cần phải thực hiện bất kỳ sửa đổi nào trên hệ thống dữ liệu gốc.

Trong công nghệ chăm sóc y tế và đặc biệt đối với các hệ thống ảnh tư liệu, mô hình giấu tin trong ảnh rất phù hợp trong việc ứng dụng kho dữ liệu Data warehouse. Ở đây, người ta phân chia dữ liệu ảnh từ tiêu đề như là tên của bệnh nhân, ngày, và tên bác sĩ theo dõi vv… Có những lúc đường link giữa ảnh và bệnh nhân không kết nối được. Do đó, việc nhúng (embedding) tên bệnh nhân vào trong ảnh có độ an toàn rất cao [1, 6]. Một câu hỏi đặt ra là việc nén dữ liệu có làm ảnh hưởng đến độ chính xác của kết quả chẩn đoán từ bác sĩ hay không? Câu trả lời là không. Một vấn đề khác nổi lên liên quan đến công nghệ chăm sóc sức khỏe là giấu các thông điệp vào trong các dãy DNA [14]. Điều này thường được dùng để bảo vệ

cái gọi là trí tuệ trong y học như là sinh học phân tử hay được gọi là các Gen (genetics). Hệ thống giấu tin mật cũng rất có ý nghĩa đối với Data warehouse, một trong hệ cơ sở dữ liệu chuyên ngành phục vụ nhiều mục đích khác nhau trong đó có An ninh Quốc gia.

Để ứng dụng Data warehouse vào trong lĩnh vực y tế của ngành Công an, chúng ta cần bổ sung thêm phần an toàn – bảo mật thông tin cho hệ thống. Do đặc thù của ngành y tế trong lực lượng vũ trang nói chung và trong ngành y tế Công an nói riêng. Chương 2 sẽ giải quyết vấn đề này.

Chương 2 - XÂY DỰNG THUẬT TOÁN GIẤU THÔNG TIN MẬT TRONG CƠ SỞ DỮ LIỆU DATA WAREHOUSE

Ta hình dung Data warehouse như là một rừng con của một rừng lớn, gồm các cây cần quan tâm nào đó trong một rừng. Mỗi "cây" trong Data warehouse có những đặc tính chung nào đó nhưng chúng có những đặc điểm riêng biệt cần được giữ bí mật. Như vậy, mỗi cây trong Data warehouse được coi như là một thông báo (message) trong đó có những dữ liệu hoàn toàn được công khai nhưng cũng có những dữ liệu có tính riêng tư cần phải giữ bí mật. Vấn đề đặt ra, làm sao tất cả đều được công khai mà vẫn giữ được bí mật riêng tư?

Có thể có một số phương pháp khác nhau để giải quyết bài toán đó, trong phạm vi nghiên cứu của luận văn, học viên đề xuất một phương pháp khá đơn giản đó là: giấu dữ liệu cần giữ bí mật vào trong ảnh số. Như vậy mỗi cây trong Data warehouse là một thông điệp chứa ảnh số. Từ đó, hình thành một hệ thống Data warehouse gồm các ảnh số cùng các quan hệ giữa chúng.

Bài toán đặt ra là xây dựng một hệ thống giấu tin mật trong ảnh sao cho phần bảo mật được bảo vệ an toàn.

2.1. Tổng quan về giấu tin

2.1.1. Khái niệm giấu tin

"Giấu tin" là một kỹ thuật nhúng (*Embedding*) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác. Kỹ thuật giấu tin nhằm hai mục đích: Một là, bảo mật cho dữ liệu được đem giấu. Hai là, bảo vệ cho chính đối tượng mang tin giấu. Hai mục đích khác nhau này dẫn đến hai kỹ thuật chủ yếu của giấu tin. Đó là giấu tin mật (*Steganography*) và thủy vân số (*Watermarking*).

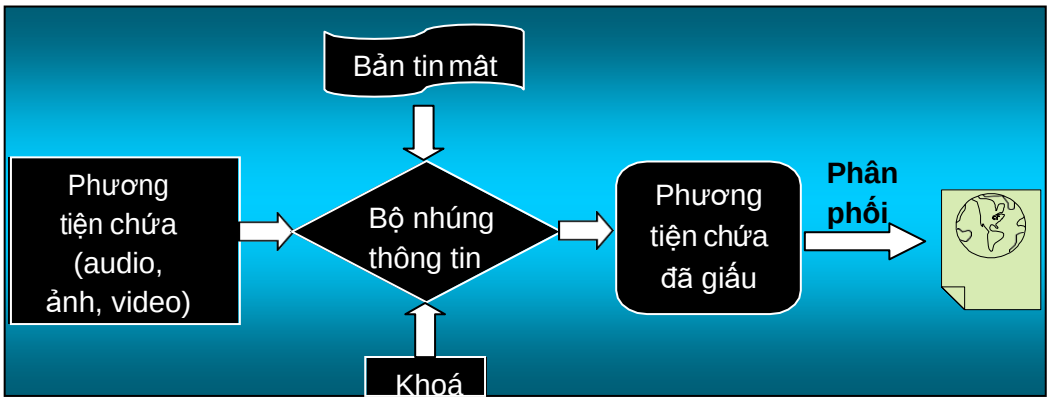
- Kỹ thuật giấu tin mật (*Steganography*): Với mục đích đảm bảo an toàn và bảo mật thông tin được giấu. Các kỹ thuật giấu tin mật tập trung vào việc sao cho thông tin giấu được nhiều và người khác khó phát hiện ra thông tin có được giấu bên trong hay không.

- Kỹ thuật thủy vân số (*Watermarking*): Với mục đích bảo mật cho chính các đối tượng giấu tin. Đảm bảo một số các yêu cầu như: tính bền vững, khẳng định bản quyền sở hữu hay phát hiện xuyên tạc thông tin…

Môi trường giấu tin bao gồm giấu tin trong ảnh, trong audio, trong video, trong văn bản dạng text… Để thực hiện giấu tin, cần xây dựng được các thủ tục giấu tin. Các thủ tục này sẽ thực hiện việc nhúng thông tin cần giấu vào môi trường giấu tin. Các thủ tục giấu tin thường được thực hiện với một khóa giống như các hệ mật mã để tăng tính bảo mật. Sau khi giấu tin ta thu được đối tượng chứa thông tin giấu (*Steganography*) và có thể phân phối đối tượng đó trên kênh thông tin.

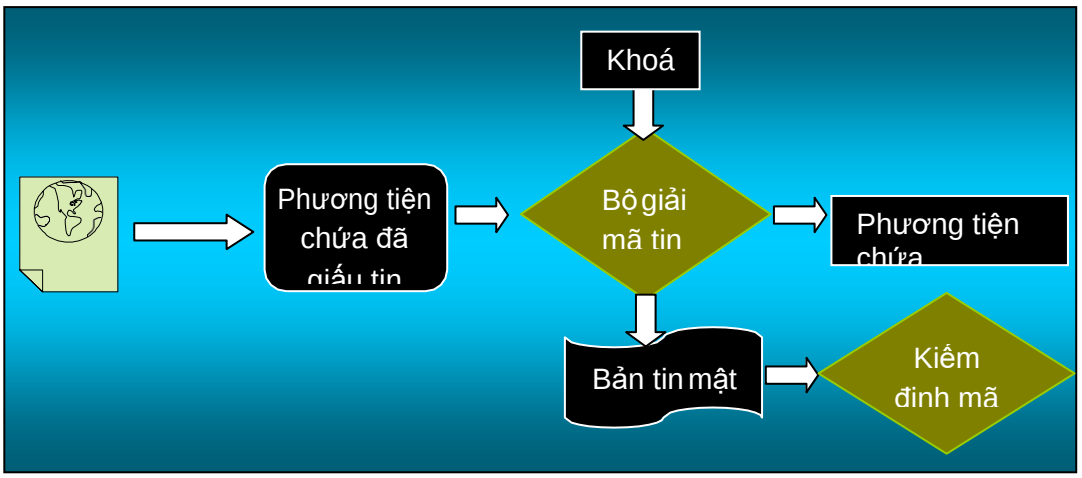
Giấu thông tin vào phương tiện chứa và tách lấy thông tin là hai quá trình trái ngược nhau và có thể mô tả qua sơ đồ khối của hệ thống trong đó:

- Bản tin mật: Có thể là văn bản hoặc tệp ảnh hay bất kỳ một tệp nhị phân nào trong quá trình xử lý, chúng ta đều chuyển chúng thành chuỗi các bit.
- Phương tiện chứa: là các file ảnh, text, audio… được dùng để làm môi trường nhúng tin mật.
- Khoá bí mật K: là khoá viết mật tham gia vào quá trình giấu tin để tăng tính bảo mật.
- Bộ nhúng thông tin: là những chương trình, thuật toán nhúng tin thực hiện việc giấu tin.
- Đầu ra: là các phương tiện chứa đã có tin giấu trong đó (được gọi là Steganography).



Hình 2.1. Sơ đồ khối quá trình giấu tin

Tách thông tin từ các phương tiện chứa diễn ra theo quy trình ngược lại. với đầu ra là các thông tin đã được giấu vào phương tiện chứa. Phương tiện chứa sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.



Hình 2.2. Sơ đồ khối quá trình giải mã

Sau khi nhận được đối tượng là phương tiện chứa có giấu thông tin, quá trình giải mã được thực hiện thông qua một bộ giải mã tương ứng với bộ nhúng thông tin cùng với khoá của quá trình nhúng. Kết quả thu được gồm phương tiện chứa gốc và thông tin đã giấu. Sau đó, thông tin đã giấu sẽ được xử lý, kiểm định, so sánh với thông tin ban đầu.

2.1.2. Kỹ thuật giấu tin trong ảnh

Ngày nay, giấu tin trong ảnh chiếm tỉ lệ lớn nhất hệ thống giấu tin trong đa phương tiện. Giấu tin trong ảnh được thực hiện bằng cách thay thế một vài thông tin ít quan trọng nhất của các điểm ảnh gốc, sao cho chất lượng ảnh ít bị ảnh hưởng nhất có thể.

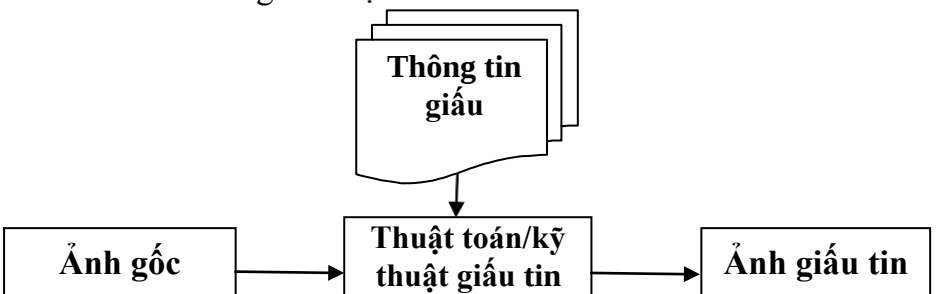
Kỹ thuật giấu tin trong ảnh bao gồm hai quá trình đó là:

- Quá trình giấu tin vào ảnh

  Đầu vào:

  - Thông tin giấu: tùy theo mục đích của người sử dụng mà giấu thông tin ở đây có thể là thông điệp, hình ảnh, video, âm thanh…
  - Ảnh gốc: là ảnh được chọn làm môi trường để giấu tin.

  Đầu ra: Ảnh đã giấu được tin.



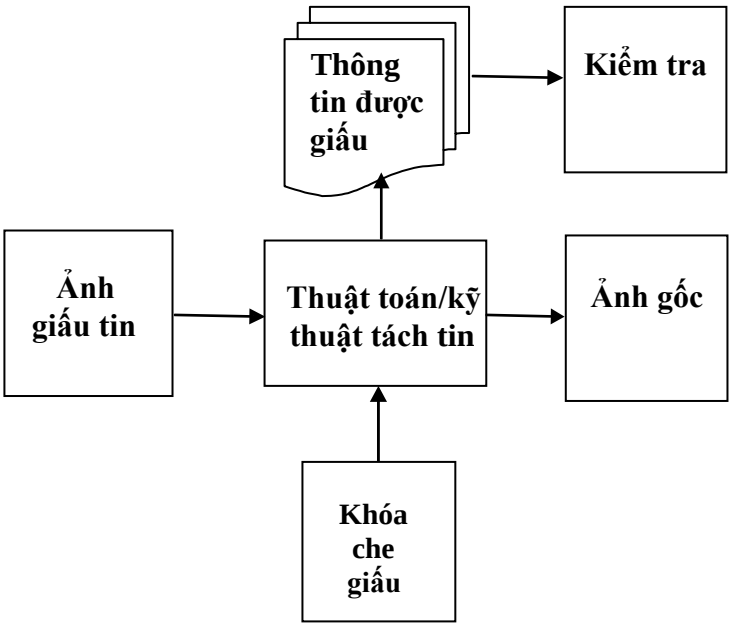
- Quá trình tách tin từ ảnh giấu tin

  Đầu vào:

  - Ảnh giấu tin.
  - Khóa che giấu.

  Đầu ra:

  - Thông tin được che giấu.
  - Ảnh đã tách tin.



Hình 2.4. Sơ đồ tách tin từ ảnh giấu tin

Quá trình giải mã được thực hiện thông qua thuật toán/kỹ thuật tách tin tương ứng với thuật toán/kỹ thuật giấu tin cùng với khóa che giấu của quá trình nhúng. Kết quả thu được gồm ảnh đã tách tin và thông tin đã giấu. Thông tin đã giấu được kiểm tra, so sánh với thông tin ban đầu. Hiện nay, đã có nhiều thuật toán giấu thông tin vào ảnh số ([5], [9], [10]). Trong phạm vi đề tài luận văn học viên trình bày một thuật toán do thầy giáo hướng dẫn cùng học viên phát triển dựa trên công cụ toán học về lý thuyết trường Galoi và mã Hamming sửa sai, sau đây là những nội dung cơ sở:

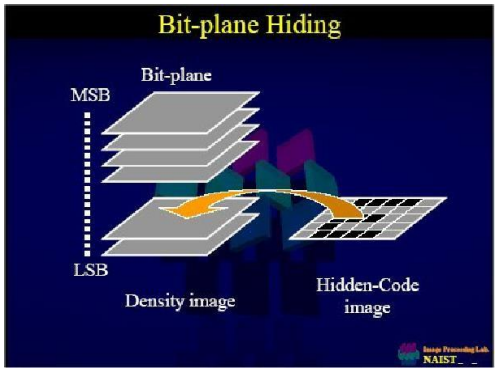
2.1.3. Kỹ thuật giấu tin mật

Trong những năm gần đây, nhiều phương pháp giấu tin mật đã được đề xuất. Phần lớn các phương pháp sử dụng đó là phương pháp thay thế. Những phương pháp này cố gắng tập trung khai thác các bít ít ý nghĩa nhất (LSB). Sự phát triển của nhiều ứng dụng đã dẫn đến các kỹ thuật xây dựng hệ thống Steganography an toàn và bền vững. Tùy theo mức độ ứng dụng, người ta chia kỹ thuật giấu tin mật thành hai lớp. Một lớp gồm các phương pháp giấu tin mật trong thông tin liên lạc và lớp còn lại liên quan đến bảo vệ bản quyền và xác thực.

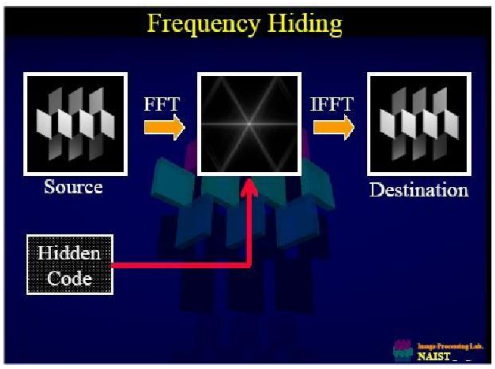
Hiện nay, đang tồn tại các phương pháp giấu tin sau đây:

+ Phương pháp thay thế: thay thế các phần dư thừa của vật mang để giấu thông điệp bí mật.
+ Dùng kỹ thuật biến đổi miền không gian: nhúng thông tin mật vào miền tần số sau khi đã biến đổi miền không gian sang miền tần số bằng các phép biến đổi toán học nào đó.
+ Các kỹ thuật trải phổ (*Spread spectrum techniques*): kỹ thuật này bắt chước ý tưởng từ thông tin liên lạc trải phổ (*Spread spectrum Communication*).
+ Kỹ thuật làm méo vật mang (*Distortion techniques*).
+ Phương pháp tạo vật mang.
+ Phương pháp thống kê.

Tuy nhiên, phương pháp được dùng phổ biến hơn cả đó là 02 phương pháp: giấu tin trong miền không gian (*Spacial Domain*) và giấu tin trong miền tần số (*Frequeney Domain*).



Hình 2.5. Giấu tin trong miền không gian



Hình 2.6. Giấu tin trong miền tần số

Hệ vật được dùng để giấu tin mật phổ biến nhất hiện nay thường là các ảnh kỹ thuật số: ảnh Bitmap và ảnh JPEG. Trong đó, giấu tin trong ảnh Bitmap (BMP) là giấu tin trong miền không gian còn giấu tin trong ảnh JPEG là giấu tin trong miền tần số - hay gọi là giấu tin trong miền biến đổi.

Ở đây, để giấu tin trong miền tần số người ta phải chuyển miền không gian về miền tần số qua phép biến đổi côsine rời rạc hoặc biến đổi furie, biến đổi wavelet, v.v. Để đơn giản trong phạm vi đề tài luận văn này học viên thực hiện việc giấu tin trong miền không gian, tức là trong ảnh Bitmap (BMP).

2.1.3.1. Kỹ thuật giấu tin với khóa bí mật

Định nghĩa 1:

Một mô hình giấu tin mật là một bộ 5 thành phần (C, M, K, E, D) ≡ Ϭ

Trong đó: C là tập các ảnh gốc dùng để giấu tin.

M là tập các thông điệp bí mật có thể thỏa mãn điều kiện /M/ ≤ /C/

K là không gian các khóa có thể.

E là không gian các ánh xạ e_k: C x M x k → C với mỗi k ϵ K.

D là không gian các ánh xạ d_k: C x k → M thỏa mãn điều kiện:

$d_k(e_k(c, m, k), k) = m$ với mỗi m ϵ M, c ϵ C, k ϵ K

Nhược điểm của mô hình này là phải giải quyết vấn đề trao đổi khóa K.

2.1.3.2. Kỹ thuật giấu tin với khóa công khai

Giống như hệ mật mã khóa công khai: khóa công khai được dùng để mã hóa thông điệp, còn khóa bí mật được dùng để giải mã. Khóa công khai được lưu trữ trên public database còn khóa bí mật được người nhận giữ bí mật, coi như khóa riêng của người nhận.

2.1.3.3. Độ an toàn của Hệ thống giấu tin mật

Một hệ thống giấu tin mật (*Steganography*) được cho là không an toàn nếu kẻ tấn công có thể chứng minh được sự tồn tại của thông điệp mật đang được lưu trong ảnh gốc (*cover image*). Vì vậy, một hệ thống Steganography được cho là an toàn nếu kẻ tấn công bằng mọi kỹ thuật và công cụ tính toán vẫn khó có thể phát hiện ra sự có mặt của thông điệp mật đang giấu trong ảnh C. Sau đây, ta sẽ mô tả độ an toàn của một hệ thống Steganography bằng các mô tả toán học.

❖ Độ an toàn hoàn hảo (Perfect Security) [10]

Cho một ảnh cover C tùy ý, được coi là một đại lượng ngẫu nhiên với phân bố xác suất $P_C(.)$. Sau khi nhúng một thông điệp m vào ảnh gốc C, ta nhận được ảnh Stego S với phân bố xác suất $P_S(.)$. Khi đó, độ đo tính không hiệu quả của thuật toán giấu tin được ký hiệu là:

$$D(P_C \| P_S) = \sum_{x \epsilon Q} P_C(x) \log_2 \frac{P_C(x)}{P_S(x)} \qquad (1)$$

Rõ ràng rằng $D(P_C \| P_S) \geq 0$ (xem [3])

Ta có định nghĩa sau:

Định nghĩa 2: Giả sử $\mathcal{S} = (C, M, K, E, D)$ là một hệ thống giấu tin mật. Khi đó, hệ thống $\mathcal{S}$ được gọi là ξ an toàn chống tại các tấn công bị động nếu:

$$D(P_C \| P_S) \leq \xi \quad (\xi \geq 0) \qquad (2)$$

Nếu $\xi = 0$ thì $\mathcal{S}$ được gọi là an toàn lý tưởng (perfectly). Trong trường hợp này, ta có: $D(P_C \| P_S) = 0$. Theo Bổ đề 1 trong [3] thì lúc đó ta có $D(P_C \| P_S) = 0 <=> P_C(.) = P_S(.)$. Do đó, kẻ tấn công bị động không thể phân biệt được đâu là ảnh C và đâu là ảnh S.

Định lý 1: Có tồn tại hệ thống Steganography an toàn hoàn hảo?

Chứng minh: Cho C là tập hợp các dãy bít có độ dài n, $P_e(.)$ có phân bố đều trên C. Ta ký hiệu e ϵ C là một thông điệp độ dài n. Người gửi A chọn một dãy c ϵ C một cách ngẫu nhiên và tính:

$$S = C \Theta e \qquad (3)$$

Ở đây phép cộng Θ ở (3) là phép XOR. Kết quả S cũng có phân bố đều trên C. Do đó, $P_S(.) = P_C(.)$. Do vậy $D(P_C \| P_S) = 0$. Đó là điều cần chứng minh.

❖ Vấn đề phát hiện ảnh chứa thông điệp mật (ảnh Stego)

Giả sử f là một ánh xạ từ C vào {0,1} được ký hiệu f : C→{0,1}, với tính chất :

$$F(e) = \begin{cases} 1 \text{ nếu C có chứa thông điệp mật} \\ 0 \text{ nếu C không chứa thông điệp mật} \end{cases} \qquad (4)$$

Người ta sử dụng hàm này để phân hoạch các ảnh trong C thành 2 lớp: lớp gồm các ảnh Stego và lớp còn lại gồm các ảnh cover. Quá trình phân lớp dựa vào hàm f (cho ở (4)) có nhiều thuật toán phân lớp các đối tượng. Song dù cho phân lớp theo thuật toàn nào cũng đều phạm phải 2 sai lầm:

- *Sai lầm loại 1* (ký hiệu là α) là sai lầm gây ra khi thực tế là ảnh C thuộc lớp thứ nhất tức là lớp các ảnh gốc nhưng ta lại gán cho nó thuộc lớp thứ 2 tức là lớp ảnh có chứa thông tin mật (gọi là xác suất bác bỏ giả thuyết đúng).

- *Sai lầm loại 2* (ký hiệu là δ) là sai lầm xảy ra khi ảnh e ϵ C có thông tin ẩn nhưng ta lại quyết định nó là ảnh cover (ảnh gốc).

Đến nay, không có một phương pháp phân lớp nào làm cực tiểu hóa cả 2 sai lầm nêu trên. Do đó, người ta muốn cố định giá trị sai lầm loại 1 α và cực tiểu hóa β. Ta có định lý sau đây:

Định lý 2: Cho Ϭ là một hệ thống Steganography thỏa mãn Ɛ an toàn. Gọi β là xác suất để kẻ tấn công không thể dò tìm ra thông điệp chứa trong ảnh C và α là xác suất để kẻ tấn công dò tìm sai thông điệp ẩn sẽ thỏa mãn bất đẳng thức sau đây: (đặt β = 1 – δ → δ cũng bé thì β càng lớn).

Trong đó: $d(\alpha, \beta) \leq Ɛ$

$$d(\alpha, \beta) = \alpha \log_2 \frac{\alpha}{1-\beta} + (1-\alpha) \log_2 \frac{1-\alpha}{\beta} \qquad (5)$$

Đặc biệt nếu $\alpha = 0$ (gần như không có sai lầm loại 1) khi đó $\beta \geq 2^{-\varepsilon}$

Chứng minh: để chứng minh định lý 2 ta phát biểu Bổ đề sau đây.

Bổ đề 1. Cho Q_0, Q_1 là hai biến ngẫu nhiên được xác định trên tập hợp Q với phân bố xác suất lần lượt là P_{Q0}, P_{Q1}; còn f là ánh xạ từ Q vào T : $f : Q \rightarrow T$.

Khi đó: $D(P_{T0} \| P_{T1}) \leq D(P_{Q0} \| P_{Q1})$

Trong đó: P_{T0}, P_{T1} lần lượt là phân bố xác suất của $f(Q_0)$ và $f(Q_1)$, chứng minh (xem [10], pp 26-28).

Bây giờ ta chứng minh định lý 2.

Xét biến ngẫu nhiên f(C) được xác định trong (4) và ký hiệu phân bố xác suất của f(e) là H_C. Trường hợp f(C) = 1. Kẻ tấn công mắc sai lầm loại I. Do đó $J_C(1) = \alpha$ và $J_C(0) = 1-\alpha$; Nếu ảnh C không chứa thông điệp mật thì ảnh cover có phân bố xác suất P_S. Ta lại tính phân bố xác suất của f(S) là J_S. Trong trường hợp đó, f(S) = 0, kẻ tấn công mắc sai lầm loại II vì anh ta không dò tìm được thông điệp ẩn trong đó, do vậy $J_S(0) = \beta$ và $J_S(1) = 1-\beta$. Vì vậy:

$$D(J_C \| J_S) = \sum_{q \in \{0,1\}} J_C(q) \log_2 \frac{J_C(q)}{J_S(q)} = (1-\alpha)\log_2 \frac{1-\alpha}{\beta} + \alpha \log_2 \frac{\alpha}{1-\beta} = d(\alpha,\beta) \quad (6)$$

Áp dụng Bổ đề 1. Ta nhận được kết quả $d(\alpha,\beta) = D(J_C \| J_S) \leq D(P_C \| P_S) \leq \varepsilon$

Đó là điều cần chứng minh.

Chú ý: 1) Trường hợp $\alpha \rightarrow 0$, ta dùng quy tắc De Loptals' ta có

$$\lim_{\alpha \rightarrow 0} \alpha \log \frac{\alpha}{1-\beta} = 0 \quad (0 < \beta < 1). \text{ Do đó } d(0,\beta) = \log_2 \frac{1}{\beta} \leq \varepsilon$$

Hay $\beta \geq 2^{-\varepsilon}$

2) $\beta = 1-\delta$ trong đó δ là xác suất sai lầm loại II

Ví dụ: 1) Cho $\varepsilon = 0{,}01$. Khi đó $\beta = 0{,}93 = 93\%$, tức là xác suất sai lầm loại II $\delta = 0{,}03 = 3\%$

2) Khi $\varepsilon = 0{,}001$ thì $\beta \geq 99\%$, tức là xác suất sai lầm loại II là 1%.

Định lý 2 là cơ sở để đánh giá chất lượng của các thuật toán giấu tin mật.

2.1.4. Kỹ thuật giấu tin mật trong các tệp ảnh Bitmap

Đến nay, đã có nhiều thuật toán giấu tin ẩn vào các tệp ảnh, nhưng phổ biến nhất hiện nay đang được ứng dụng rộng rãi trên toàn thế giới là kỹ thuật chèn các bít thông tin ẩn vào các bít có ý nghĩa thấp nhất (Least signifieant Bit – LSB) trong phần dữ liệu ảnh của ảnh Bmp 24 bít màu. Do việc thay đổi các bít LSB chỉ gây ra sự thay đổi nhỏ nhất của các thành phần màu mà mắt thường không thể nhận biết được sự thay đổi đó. Vậy các tham số cần tính toán khi áp dụng kỹ thuật chèn bít LSB là gì ?

Thứ nhất, kích cỡ dữ liệu mật cần giấu: khi muốn ẩn một đoạn văn bản hay một file dữ liệu ẩn bất kỳ vào một file ảnh Bmp nào đó chúng ta cần đảm bảo rằng, chất lượng và kích cỡ của file ảnh đó không bị thay đổi. Vì vậy, độ dài tối đa của đoạn văn hoặc tệp dữ liệu ẩn đó (tính ra bít) là bao nhiêu để đạt được yêu cầu đặt ra? Với phương pháp LSB người ta đã tính được độ dài tối đa của dữ liệu ẩn so với độ dài các LSB của dữ liệu ảnh là:

$$L_{max} \approx 12{,}5\%\ Lbmp$$

Trong đó: Lbmp là độ dài của một file ảnh Bitmap mang thông tin. Tức là không được giấu quá 1 bít dữ liệu ẩn vào một điểm ảnh (pixel).

Thứ hai, xác định vị trí đặt dữ liệu ẩn đầu tiên (gọi là khởi điểm giấu tin): Khi muốn đặt các bít thông tin ẩn vào một file ảnh Bitmap, thì vấn đề đầu tiên là phải xem đặt thông tin ẩn bắt đầu từ vị trí nào của file ảnh là tốt nhất? Chúng ta biết rằng 32 bytes (256 bits) đầu tiên của một file ảnh Bitmap được dành cho cấu trúc Header của nó. Tiếp theo cấu trúc Header này là các thông tin về bảng màu của ảnh mà chúng ta gọi chung các thành phần này là phần chứa thông tin về ảnh, và do vậy chúng ta không được chèn thông tin ẩn vào đó. Độ dài của phần chứa thông tin về ảnh nằm ở byte thứ 11 sau cấu trúc Header. Do đó, dữ liệu ẩn chỉ được phép chèn vào phần dữ liệu ảnh của file ảnh Bitmap. Hơn nữa, nếu thông tin ẩn được chèn bắt đầu vào vị trí từ Byte đầu tiên của phần dữ liệu ảnh thì tính bảo mật không cao. Vì vậy, để tăng độ bảo mật cho dữ liệu ẩn thì dữ liệu ẩn nên được chèn bắt đầu từ phần dữ liệu ảnh tại một vị trí ngẫu nhiên liên quan đến mật khẩu:

$$Offset = f(c_1, c_2, \ldots, c_n)$$

Trong đó: $c_1, c_2, \ldots, c_n$ là dãy con của dãy ký tự mật khẩu độ dài N tức là

$n = 1, 2, \ldots, N$.

Ví dụ: nếu N = 4 thì PSW[4] là một mảng cấu trúc 4 phần tử, với mỗi phần tử là 1 ký tự tự nhiên.

Thứ ba, mã hóa thông điệp trước lúc nhúng vào ảnh môi trường. Điều này đảm bảo an toàn cho những thông điệp cần có độ bảo mật cao nhất để đề phòng đối phương có thể phát hiện và trích chọn được thông điệp trong ảnh Stego.

2.1.4.1. Giới thiệu về cấu trúc của ảnh Bitmap (BMP)

Ảnh Bitmap được phát triển đầu tiên bởi Microsoft Corporation và được lưu trữ dưới dạng độc lập với các thiết bị lưu, cho phép Windows hiển thị dữ liệu không phụ thuộc vào khung chỉ định màu trên bất kỳ phần cứng nào. Tệp mở rộng mặc định của một file ảnh Bitmap là BMP. Ảnh BMP được sử dụng rộng rãi trên môi trường Microsoft Windows và cài ứng dụng chạy trên môi trường Windows từ phiên bản 3.0 trở lên. Nói tóm lại, một file ảnh BMP gồm 3 phần:

- Bitmap Header (tiêu đề ảnh);
- Palette (bảng màu);
- Bitmap Data (dữ liệu ảnh).

2.1.4.2. Giấu tin trong ảnh màu và đa cấp xám

Ta biết rằng, việc giấu tin trong ảnh đen-trắng đem lại hiệu quả thấp vì việc biến đổi từ trắng sang đen (1 sang 0) hoặc từ đen sang trắng (0 sang 1) rất dễ tạo ra nhiễu cho ảnh. Vì vậy, người khác dễ phát hiện được ảnh Stego bằng mắt thường. Hơn nữa, việc giấu tin trong ảnh đen trắng cho tỷ lệ giấu rất thấp. Chẳng hạn, một bức ảnh đen trắng kích cỡ 3x4 chỉ có thể giấu được tối đa là khoảng 10KB dữ liệu mật trong khi đó một ảnh 24 màu với kích cỡ tương tự có thể giấu được tối đa là 200KB. Dó đó, hiện nay ảnh đen trắng ít được dùng để giấu tin, thay vào đó người ta thường dùng ảnh màu hoặc ảnh đa cấp xám làm ảnh môi trường (cover image) để giấu tin.

Để lựa chọn ảnh màu và đa cấp xám làm ảnh môi trường cho việc giấu tin mật, chúng ta quan tâm đến bít ít có ý nghĩa nhất (Least Signifieant Bit) mà ta ký hiệu là LSB. Đó là bít có ít ảnh hưởng nhất đến màu sắc của ảnh. Do vậy, khi mỗi LSB bị thay đổi thì màu sắc của ảnh không thay đổi đáng kể so với ảnh gốc ban đầu [10]. Nhưng làm thế nào để xác định được đâu là LSB của mỗi điểm ảnh? Việc xác định LSB của mỗi điểm ảnh (pixel) của một bức ảnh phụ thuộc vào định dạng của ảnh đó và số bít màu dành cho mỗi điểm ảnh.

Quá trình giấu thông tin trong ảnh màu và ảnh đa cấp xám hoàn toàn tương tự như đối với ảnh đen-trắng, chúng chỉ khác nhau là ở chỗ: đối với ảnh màu và đa cấp xám, trước hết ta phải chọn từ mỗi pixel dữ liệu ảnh LSB để tạo thành một ảnh nhị phân và được gọi là ảnh thứ cấp. Ta sử dụng ảnh thứ cấp này làm ảnh môi trường để giấu tin. Sau đó, ta trả lại đúng vị trí đã trích chọn vào ảnh ban đầu để thu được ảnh kết quả được gọi là ảnh Stego.

- Đối với ảnh 16 bít màu hoặc 24 bít màu thì việc xác định LSB tương đối đơn giản. Tuy nhiên, đối với ảnh 8 bít màu trở xuống thì công việc trở nên phức tạp hơn.

- Đối với ảnh màu đa cấp xám thì bảng màu của nó đã được sắp, trong đó những cặp màu trong bảng màu có chỉ số chênh lệch càng ít thì chúng càng giống nhau.

Vì vậy, đối với loại ảnh này thì LSB của mỗi pixel dữ liệu ảnh là bít cuối cùng của mỗi pixel dữ liệu ảnh. Quá trình tách LSB của ảnh đa cấp xám và thay đổi các bít này bằng thuật toán như thuật toán giấu tin trong ảnh đen trắng sẽ làm cho chỉ số màu của mỗi điểm màu thay đổi tăng hoặc giảm đi chỉ một đơn vị. Do đó, điểm ảnh mới sẽ có độ sáng/tối của ô màu liền trước hoặc sau ô màu của điểm ảnh cũ và bằng mắt thường khó có thể phát hiện được sự thay đổi đó. Thực nghiệm đã chỉ ra rằng: ngay cả khi ta đảo toàn bộ các LSB của từng điểm ảnh trong một ảnh 8 bít đa cấp xám thì cũng không thể hiện sự khác nhau đáng kể.

Trường hợp ảnh nhỏ hơn hoặc bằng 8 bít màu, những ảnh thuộc loại này bao gồm ảnh 16 màu (4 bít màu) và ảnh 256 màu (gồm 8 bít màu). Khác với ảnh màu đa cấp xám, ảnh màu với 4 bít màu hoặc 8 bít màu không phải luôn luôn được sắp xếp bảng màu: những màu liền kề nhau (trong bảng màu) có thể rất khác nhau. Chẳng

hạn, màu đen và màu trắng có thể được sắp xếp kề nhau. Do đó, việc xác định LSB là một khó khăn. Nếu chúng ta cứ làm như đối với ảnh đa cấp xám, nghĩa là vẫn lấy bít cuối cùng của mỗi điểm ảnh để tạo thành ảnh thứ cấp thì mỗi khi thay đổi 0 sang 1 (hoặc ngược lại, 1 sang 0) trên ảnh thứ cấp thì có thể làm cho màu của điểm ảnh gốc và của ảnh Stego sẽ khác nhau rất nhiều mà mắt thường có thể nhận biết được mặc dù màu của chúng cũng chỉ tăng (hoặc giảm) một bít mà thôi.

Để gải quyết vấn đề này, chúng ta cần sắp xếp lại thứ tự các điểm ảnh trong bảng màu của ảnh gốc sao cho 2 màu kề nhau sẽ khác nhau ít nhất bằng cách so sánh thành phần màu của từng cặp điểm với nhau. Tuy nhiên, màu của một điểm ảnh làm một vectơ 3 thành phần không sắp thứ tự nên việc so sánh chỉ có tính tương đối mà thôi. Quá trình sắp xếp lại bảng màu được thực hiện như sau:

Step 1. Chọn một màu bất kỳ trong bảng màu, giả sử đó là màu A(x, y, z) để đưa vào vị trí đầu tiên trong bảng màu, trong đó ký hiệu x là thành phần Red, y là thành phàn Green và z là thành phần Blue;

Step 2. Chọn màu $B(m_0, n_0, p_0)$ sao cho: $S(A, B(m_0, n_0, p_0)) = S(A, B) = \min \text{Sqrt}[(x - m)^2 + (y - n) + (z - p)^2]$;

Step 3. Xếp màu $B(m_0, n_0, p_0)$ cạnh màu A(x, y, z);

Step 4. Quay lại bước 2 cho đến khi tất cả các màu của bảng màu đã được sắp và quá trình kết thúc.

Lưu ý:

a) *Đối với ảnh 16 bít màu*: Trong thực tế chỉ dùng đến 15 bít cho mỗi điểm ảnh, trong đó 5 bít biểu diễn cường độ cho màu Red; 5 bit biểu diễn cường độ cho màu Green và 5 bít còn lại biểu diễn cường độ cho màu Blue; còn 1 bít cuối cùng không được dùng đến. Đó chính là bít thấp nhất của Byte thứ 2 trong một cặp 2 Byte để biểu diễn một điểm ảnh. Do là, LSB của ảnh 16 bít màu.

b) *Đối với ảnh 24 bít màu*: Sử dụng 3 Byte cho mỗi điểm ảnh. Trong mỗi Byte, bít càng thấp càng ít ảnh hưởng đến màu của mỗi điểm ảnh. Vì vậy, bít cuối cùng của mỗi Byte trong phần dữ liệu ảnh là các LSB của ảnh. Như vậy, cứ một điểm ảnh ta có thể trích chọn được 03 bít LSB để thực hiện việc giấu tin. Thậm chí

ta có thể sử dụng được nhiều hơn thế để tạo tành ảnh thứ cấp cho việc giấu tin mà không ảnh hưởng đáng kể đến chất lượng của ảnh gốc. Khi đó, lượng thông tin giấu được là rất lớn, có thể đạt tỷ lệ 1/8 ≈ 12,5% so với kích cỡ của ảnh gốc. Để đảm bảo độ an toàn cao nhất có thể, chúng ta có thể nén (zip) bản thông điệp sau đó mã hóa kết quả rồi đem giấu vào ảnh.

2.2. Cơ sở toán học xây dựng thuật toán

Trước hết ta ký hiệu: GF(P) là trường Galois có cấp là số nguyên tố P còn GF(P)[x] là không gian véctơ các đa thức với các hệ số trên trường GF(P). Khái niệm đa thức nguyên thủy và ứng dụng:

2.2.1. Định nghĩa 1

Đa thức f(x) có cấp m trong trường GF(p) được gọi là đa thức bất khả qui (*irredueible polynomial*) nếu f(x) không thể được phân tích thành tích của các đa thức có cấp nhỏ hơn m (và > 1) trong trường GF(q).

Ví dụ: $f(x) = x^2 + x + 1$; $f(x) = x^{11} + x^2 + 1$ đều là những đa thức bất khả qui trong trường GF(2)

2.2.2. Định nghĩa 2

Đa thức bất khả qui p(x) cấp m được gọi là đa thức nguyên thủy (primitive polynomial) trong trường GF(P) nếu số nguyên dương nhỏ nhất n mà $x^n - 1$ chia hết cho P(x) phải thỏa mãn $n = P^m - 1$

Ví dụ: đa thức bất khả qui $P(x) = x^3 + x + 1$ là đa thức cấp 3 trong trường GF(2) là đa thức nguyên thủy trong trường GF(2) vì số n bé nhất là $n = 2^3 - 1 = 7$ mà $x^7 - 1$ chia hết cho P(x). Thật vậy, $x^7 - 1 = (x^3 + x + 1)(x^4 + x^2 + x + 1)$ và mọi số $n < 7$ thì $x^n - 1$ không thể chia hết cho $x^3 + x + 1$

Trong phạm vi đề tài luận văn này, học viên chỉ xét các đa thức trong trường GF(2):

- Định lý 1. Có tất cả Ø $(2^m - 1)/m$ đa thức nguyên thủy cấp m trong trường GF(2). Trong đó, Ø (.) là hàm phi-ơle.
- Định lý 2. Tập các nghiệm $\{\alpha_i\}$ của đa thức nguyên thủy P(x) cấp m trong trường GF(2) sẽ có cấp $2^m - 1$.

Chứng minh (xem [7])

Bây giờ ta giả sử α là một nghiệm của đa thức nguyên thủy P(x) có cấp m là:

$P(x) = x^m + a_{m-1}x^{m-1} + \ldots + a_1x + a_0$ (chú ý a_0 luôn khác không) với $a_i \in GF(2)$

Khi đó $P(\alpha) = \alpha^m + a_{m-1}\alpha^{m-1} + \ldots + a_1\alpha + a_0 = 0$

Từ đây, ta suy ra: $\alpha^m = -a_0 - a_1\alpha - \ldots a_{m-1}\alpha^{m-1}$ [8]

Do $a_i \in GF(2) = \{0,1\}$ với mọi i = 0, 1, …, m-1. Trong trường GF(2) thì a + b = a – b nên ta có thể viết biểu thức cho ở [8] là: $\alpha^m = a_0 + a_1\alpha + \ldots + a_{m-1}\alpha^{m-1}$ [9]

Do định lý 2, ta suy ra: các lũy thừa của α có cấp lớn hơn hoặc bằng m có thể được biểu diễn dưới dạng đa thức có cấp nhỏ hơn m. Vì α có cấp $2^m - 1$ nên các lũy thừa khác nhau của α phải có $2^m - 1$ các biểu diễn đa thức phân biệt khác không dưới dạng: $P(x) = b_0 + b_1\alpha + \ldots + b_{m-1}\alpha^{m-1}$ với $b_i \in GF(2)$, i = 0, 1, …, m-1. Như vậy, tập hợp tất cả các nghiệm của đa thức nguyên thủy cấp m cùng với véctơ 0 có thể được xem như là một không gian véctơ trên trường GF(2).

Bây giờ ta lấy m = 6. Khi đó không gian véctơ $GF(2^6)$ trên trường GF(2) chúng gồm 64 phần tử (kể cả véctơ 0). Mỗi phần từ là một véctơ gồm 6 thành phần nhị phân. Để ứng dụng cho thuật toán giấu tin mật, ta chọn một đa thức nguyên thủy cấp 6 trong trường GF(2).

Dễ thấy rằng $P(x) = x^6 + x + 1$ là một đa thức nguyên thủy trong trường GF(2).

Vì véctơ (0, 0, 0, 0, 0, 0) không phải là một nghiệm của đa thức $P(x) = x^6 + x + 1$, nên ta sẽ có $2^6 - 1$ nghiệm của P(x).

Tất cả 63 nghiệm của $x^6 + x + 1$ được lập như sau: Lấy α là một nghiệm tùy ý của P(x), ta có $\alpha^6 + \alpha + 1 = 0$.

Do đó: $\alpha^6 = \alpha + 1$; $\alpha^7 = \alpha^2 + \alpha$; $\alpha^8 = \alpha^3 + \alpha^2$; $\alpha^9 = \alpha^4 + \alpha^3$; $\alpha^{10} = \alpha^5 + \alpha^4$; $\alpha^{11} = \alpha^6 + \alpha^5$; $\alpha^{12} = \alpha^7 + \alpha^6$; v.v.

Trong không gian véctơ $GF(2^6 - 1)$ có 6 chiều nên nó có cơ sở trực chuẩn gồm 6 véctơ cực đại độc lập tuyến tính. 6 véctơ của cơ sở này được ký hiệu là:

S = {100000, 010000, 001000, 000100, 000010, 000001}

Tất cả véctơ đó trong $GF(2^6 – 1)$ trừ véc tơ 0 = (000000) được thiết lập bởi bảng H_{6x63} như sau:

$$H = \begin{pmatrix} 1000001000011000101001111010001110010010110111011001101010111111 \\ 0100001100010100111101000111001001011011101100110101011111110000 \\ 0010000110001010011110100011100100101101110110011010101111111000 \\ 0001000011000101001111010001110010010110111011001101010111111100 \\ 0000100001100010100111101000111001001011011101100110101011111110 \\ 0000010000110001010011110100011100100101101110110011010101111111 \end{pmatrix}$$

$$\Rightarrow H = [h_{.1}, h_{.2}, \ldots, h_{.63}] \text{ với } h_{ij} \in \{0,1\} \text{ và } h_{.j} = (h_{1j}, h_{2j}, h_{3j}, \ldots, h_{6j})^T \quad j = \overline{1,63}$$

Kết quả chính đạt được ở chương 2 là tiền đề để xây dựng một thuật toán giấu tin mật có độ an toàn cao, cho phép cải tiến để có thể giấu được mọi thông điệp mật. Thuật toán được đề xuất có thể giấu thông điệp vào trong ảnh Bmp, mầu đa cấp xám.

Việc giấu tin trong miền tần số đòi hỏi phải chuyển miền không gian về miền tần số bởi phép biến đổi cosine rời rạc hoặc phép biến đổi Furie rời rạc hoặc biến đổi Wavelet. Luận văn này không đề cập tới vấn đề này. Mặt khác, để tăng cường độ an toàn cho thông điệp ta có thể dùng thuật toán mã hóa khóa đối xứng AES sau đó giấu bảng mã vào trong ảnh.

Tuy nhiên, trong phạm vi đề tài luận văn, học viên không đi sâu vào vấn đề này vì ngoài mã hóa còn liên quan đến các vấn đề trao đổi khóa mã khá phức tạp. Do vậy, học viên xin được bỏ qua và sẽ tiếp tục nghiên cứu sau.

Chương 3 – ĐỀ XUẤT THUẬT TOÁN GIẤU TIN MẬT VÀ ỨNG DỤNG TRONG NGÀNH Y TẾ

"*Chương trình Chuyển đổi số quốc gia đến năm 2025, định hướng đến năm 2030*" mới được Thủ tướng Chính phủ phê duyệt có 8 ngành, lĩnh vực cần được ưu tiên chuyển đổi số trước bao gồm: Y tế, Giáo dục, Tài chính – Ngân hàng, Nông nghiệp, Giao thông vận tải và logistic, Năng lượng, Tài nguyên và Môi trường, Sản xuất công nghiệp. Đây là những lĩnh vực có tác động trực tiếp đến xã hội, liên quan hàng ngày tới người dân.

Chuyển đổi số trong lĩnh vực y tế sẽ tập trung vào các nhiệm vụ như: phát triển nền tảng hỗ trợ khám, chữa bệnh từ xa, xây dựng hệ thống chăm sóc sức khỏe và phòng bệnh dựa trên các công nghệ số, sử dụng hồ sơ bệnh án điện tử, thanh toán viện phí, hình thành các bệnh viện thông minh. Xây dựng nền tảng quản trị y tế thông minh dựa trên công nghệ số, tích hợp thông tin, dữ liệu, hình thành cơ sở dữ liệu quốc gia về y tế. Đồng thời, tạo hành lang pháp lý cho khám chữa bệnh từ xa và đơn thuốc điện tử nhằm bảo đảm người dân có thể tiếp xúc với bác sĩ nhanh chóng, hiệu quả, giảm chi phí và thời gian vận chuyển bệnh nhân...

Trong ngành y tế Công an, khi ứng dụng công nghệ thông tin đáp ứng chuyển đổi số cũng đặt ra các yêu cầu bảo mật dữ liệu trong quản lý, lưu trữ dữ liệu và trao đổi thông tin qua mạng máy tính. Đặc biệt, cần bảo đảm tính bí mật một số thông tin cá nhân, do vậy việc mã hóa các thông tin cá nhân người bệnh trong ảnh số cũng là một trong những nhu cầu thực tiễn hiện nay.

Dựa trên các tìm hiểu tại Chương 1, Chương 2 học viên xin đề xuất thuật toán giấu tin mật trong cơ sở dữ liệu Data warehouse làm tiền đề tiến tới xây dựng chương trình phần mềm tìm kiếm các thông tin liên quan đến Bảo hiểm y tế nhằm hiện thực hóa các nhu cầu quản lý, chuyển đổi số trong lĩnh vực y tế như đã phân tích ở trên.

3.1. Thuật toán giấu tin và trích chọn tin mật

3.1.1. Thuật toán Giấu tin mật (embed)

Trên cơ sở ma trận H đã được xây dựng ở chương 2, ta đề xuất thuật toán giấu tin mật như sau:

Input: Bản thông điệp M = (m_1 m_2 m_3… m_n), $m_i \in$ {a, b, c, …, z}={0, 1, 2, …, 25}; Ảnh Bitmap C; khởi điểm giấu (điểm bắt đầu đặt dữ liệu vào ảnh C)

Output: Ảnh Stego S.

Bước 1. Dùng thuật toán nén Zip để nén bản thông điệm M, ta được Zip (M) = X = (x_1, x_2, …, x_k);

Bước 2. Chuyển dãy ký tự của X thành dãy nhị phân và phần kết quả được chia thàng từng block có độ dài bằng nhau và bằng 6 (nếu khối cuối cùng không bằng 6 thì thêm vào các số 0 cho đủ 6 bit); kết quả được ký hiệu là Y = (y_1, y_2, …, y_n);

Bước 3. Trích chọn 63n các LSB của các pixel dữ liệu ảnh của C bắt đầu từ khởi điểm cho trước, ta được:

Z = (z_1, z_2, …, z_n), mỗi Z_i = (z_{i1} z_{i2} … z_{i63}) i = 1, 2, …, n. $z_{ij} \in \{0,1\}$;

Bước 4. Với i = 1, 2, …, n, tính:

$u_i = y_i \Theta Hz_i$ (where x^T là chuyển vị của véctơ x) và phép toán Θ là phép cộng XOR;

Bước 5. Với mỗi i tìm trong ma trận H có cột $h_{.i}$ nào trùng với u_i^T hay không nếu không tìm thấy thì cho qua và vectơ y_i được giữ nguyên;

Bước 6. Giả sử có tồn tại một i mà $h_{.i}$ trùng với u_i^T thì ta đảo bit thứ j của véctơ z_i tại bít z_{ij} tạo thành bit $z_{ij}' = z_{ij} \Theta 1$ để nhận được véctơ $z_i' = (z_{i1}, …, z_{ij}', z_{ij+1}, …, z_{i63})$ và quay lại Bước 4;

Bước 7. Trả lại tất cả các bit $z' = (z_1', z_2', …, z_n')$ lần lượt đúng như vị trí đã trích chọn z (tức là thay z' bằng z vào ảnh C) ta nhận được ảnh mới S.

3.1.2. Thuật toán Trích chọn (extract)

Input: Ảnh S và khởi điểm giấu.

Output: Thông điệp mật M.

Bước 1. Trích chọn được $(z'_1, z'_2, \ldots, z'_n) = z'$ với $z'_i = (z'_{i1}, z'_{i2}, \ldots, z'_{i63})$ i = 1, 2, …, n; như vậy: $z' = (z'_1, z'_2, \ldots, z'_{63n})$.

Bước 2. Chia dãy z thành từng block (mỗi block có độ dài 6) ta nhận được kết quả:

$Y = (Y_1, Y_2, \ldots, Y_n)$, $Y_i = (y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5}, y_{i6})$ i = 1, 2, …n;

Bước 3. For i = 1, 2, …, n. Tính $X_i^T = HY_i^T$ ta được véctơ $X = (X_1, X_2, \ldots, X_n)$;

Bước 4. Giải nén Zip (unzip): M = unzip (X) là thông điệp được giấu trong ảnh S.

Chú ý: muốn khôi phục lại ảnh gốc C ta chỉ cần giấu trở lại thông điệp M đã được trích chọn bắt đầu từ khởi điểm đã cho.

3.1.3. Phạm vi ứng dụng và lý do sử dụng thuật toán

Học viên đề xuất thuật toán trên nhằm mục đích mã hóa thông điệp 6 bit vào ảnh số thẻ BHYT. Khi sử dụng thuật toán này số lượng ký tự được mã hóa đã tăng lên từ 31 ký tự (mã hóa 5 bít) thành 63 ký tự (mã hóa 6 bít) đáp ứng yêu cầu thông tin cần giấu. Từ ma trận H với 63 mã nhị phân (trang 30) ta xây dựng được bảng mã tương ứng với chữ số, chữ cái la tinh cụ thể như sau:

STT	Ký tự	Từ mã
1		000000
2	a	010000
3	b	001000
4	c	000100
5	d	000010
6	e	000001
7	f	010100
8	g	001010
9	h	000101
10	i	010110
11	j	001011
12	k	010001
13	l	011100
14	m	001110
15	n	000111
16	o	010111
17	p	011111
18	q	011011
19	r	011001
20	s	011000
21	t	001100

STT	Ký tự	Từ mã
22	u	000110
23	v	000011
24	w	010101
25	x	011110
26	y	001111
27	z	011101
28	/	011010
29	0	100000
30	1	110000
31	2	111000
32	3	111100
33	4	111110
34	5	111111
35	6	101000
36	7	101100
37	8	101110
38	9	101111
39	A	100100
40	B	100110
41	C	100111
42	D	100010

STT	Ký tự	Từ mã
43	E	100011
44	F	100001
45	G	101010
46	H	101011
47	I	100101
48	J	110001
49	K	110010
50	L	110111
51	M	110011
52	N	101001
53	O	110100
54	P	111101
55	Q	111011
56	R	101101
57	S	111001
58	T	110101
59	U	111010
60	V	010011
61	W	001001
62	Y	010010
63	Z	001101

3.1.4. Thử nghiệm và đánh giá thuật toán

3.1.4.1. Thử nghiệm

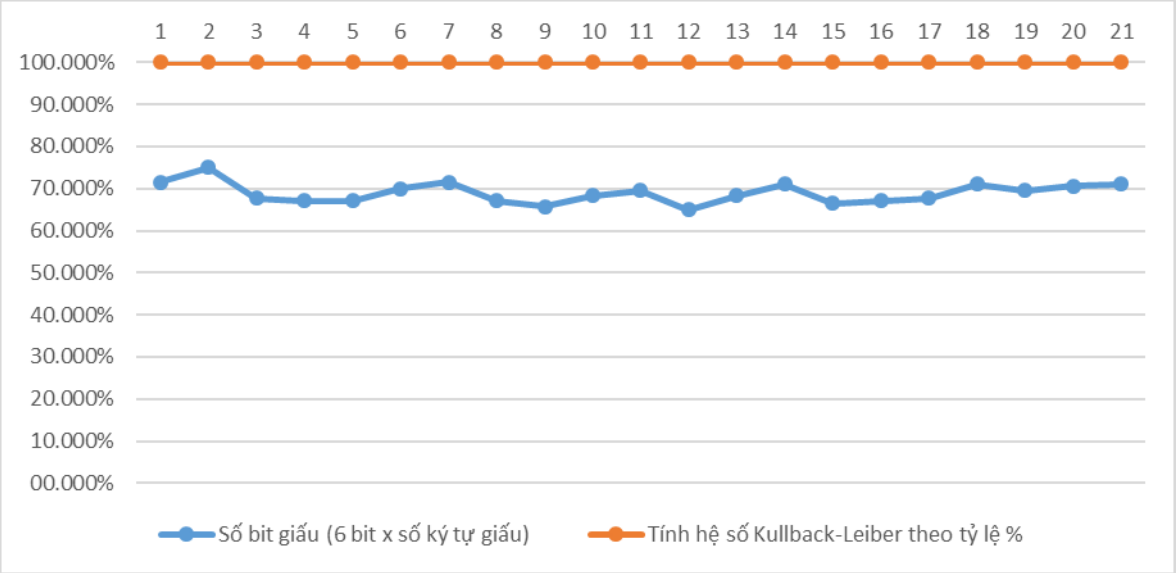
Để so sánh hiệu quả giữa thuật toán được đề xuất ở trên và thuật toán đã được công bố trong [9], học viên sử dụng máy tính cấu hinh CPU Intel core i5-6200U, 2.3Ghz 8Gb RAM. Thuật toán được mô phỏng thực nghiệm bởi ngôn ngữ Python.

Hình 3.1. Ứng dụng mã hóa thông điệp 6 bits vào ảnh

Kết quả thực nghiệm trên 21 ảnh chân dung, kích thước 4x6, có độ phân giải khoảng 300dpi, định dạng BMP được thể hiện trong bảng 3.1 dưới đây:

Bảng 3.1. Kết quả thực nghiệm

TT	Ảnh	Số bit giấu (6bit x số ký tự giấu)	Số bit ảnh gốc (427x709x24)	Xác suất thay đổi	Hệ số sai phân (Kullback-Leiber)
1	img1	252	7265832	0.00003468288284122	0.00003468348430000
2	img2	300	7265832	0.00004128914623955	0.00004128999865200
3	img3	210	7265832	0.00002890240236768	0.00002890282004600
4	img4	204	7265832	0.00002807661944289	0.00002807701359500
5	img5	204	7265832	0.00002807661944289	0.00002807701359500
6	img6	234	7265832	0.00003220553406685	0.00003220605267100
7	img7	252	7265832	0.00003468288284122	0.00003468348430000
8	img8	204	7265832	0.00002807661944289	0.00002807701359500
9	img9	192	7265832	0.00002642505359331	0.00002642540273800
10	img10	216	7265832	0.00002972818529248	0.00002972862717900
11	img11	228	7265832	0.00003137975114206	0.00003138024349200
12	img12	186	7265832	0.00002559927066852	0.00002559959833200
13	img13	216	7265832	0.00002972818529248	0.00002972862717900
14	img14	246	7265832	0.00003385709991643	0.00003385767307500
15	img15	198	7265832	0.00002725083651810	0.00002725120782500
16	img16	204	7265832	0.00002807661944289	0.00002807701359500
17	img17	210	7265832	0.00002890240236768	0.00002890282004600
18	img18	246	7265832	0.00003385709991643	0.00003385767307500
19	img19	228	7265832	0.00003137975114206	0.00003138024349200
20	img20	240	7265832	0.00003303131699164	0.00003303186253200
21	img21	246	7265832	0.00003385709991643	0.00003385767307500



Hình 3.2. Biểu đồ K-L theo số Bit giấu tin

3.1.4.2. Đánh giá kết quả đạt được

Hiệu quả của thuật toán giấu tin mật được so sánh với một số thuật toán khác đã được công bố dựa trên 03 tiêu chí:

- Tỷ lệ thông tin giấu.
- Khả năng khó có thể phát hiện thông tin ẩn trong ảnh
- Tốc độ tính toán của thuật toán.

Để đánh giá mức độ an toàn của thuật toán vừa được trình bày ở trên, người ta thường sử dụng hai tham số: Sai số bình phương trung bình – MSE (*mean square error*) và phương pháp đề xuất với hệ số tỷ lệ tín hiệu/tín hiệu tạp PSNR (*Peak Signal to Noise Ratio*).

PSNR, đơn vị: deciben (dB), thường được sử dụng trong nghiên cứu xử lý hình ảnh

$$PSNR = 10\log_{10}(MAX^2/MSE)$$

Trong đó: MAX là giá trị cực đại các pixels dữ liệu của ảnh

MSE là giá trị trung bình bình phương của hiệu $(x_{ij}-y_{ij})$ tương ứng của ảnh gốc C với ảnh Stego có chứa thông tin ẩn được tính như sau.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2$$

Ở đây: x_{ij} biểu thị giá trị điểm ảnh gốc, và y_{ij} biểu thị giá trị điểm ảnh đã được biến đổi, m và n lần lượt là chiều rộng và chiều cao của ảnh.

Ví dụ: Đối với ảnh đa cấp xám thì mỗi pixels gồm 8 bít. Vậy MAX = 2^8 -1 = 255. Như vậy, giá trị PSNR càng lớn càng tốt vì khi đó chất lượng ảnh Stego càng gần với ảnh gốc C.

Thông thường, nếu PSNR ≥ 37 dB thì hệ thống mắt người gần như không phân biệt được giữa ảnh gốc và ảnh khôi phục. PSNR càng cao thì chất lượng ảnh khôi phục càng tốt [5]. Khi hai hình ảnh giống hệt nhau, MSE sẽ bằng 0 và PSNR đi đến vô hạn.

So sánh thuật toán mã hóa thông điệp 5 bits với thuật toán mà học viên đề xuất ta có bảng kết quả 3.2, cho thấy thuật toán được đề xuất hiệu quả hơn thuật toán mã hóa thông điệp 5 bít (theo nghĩa: có độ an toàn cao hơn).

Bảng 3.2. Kết quả so sánh trên 02 thuật toán

Ảnh	PSNR (5 bit)	PSNR (6 bit)
img1	85,86165545	85,3580056
img2	84,54878612	84,72012207
img3	86,49419564	86,6731153
img4	85,48561541	85,77517996
img5	85,6803354	85,98673009
img6	85,03538319	85,8649406
img7	85,28723396	86,42363524
img8	86,23750818	86,49409422
img9	87,51817624	87,51730782
img10	85,62288864	86,50382186
img11	84,69901003	85,6830928
img12	85,63472104	86,51278891
img13	85,09351819	86,43111066
img14	85,91776568	86,96445253
img15	85,62909132	87,49821391
img16	85,41074633	86,41074956
img17	83,70750898	85,61267362
img18	85,47406771	86,76824509
img19	83,76062826	85,47582899
img20	86,27142682	86,27143321
img21	86,26959417	86,44000422

3.2. Ứng dụng trong ngành y tế

Theo thống kê, số giờ thực hiện thủ tục hành chính trong việc quản lý bảo hiểm y tế (BHYT) giảm mạnh từ 335 giờ năm 2014 xuống còn xuống còn 147 giờ năm 2018. Thời hạn cấp thẻ BHYT từ 7 ngày được rút ngắn xuống còn 5 ngày (riêng với người hưởng trợ cấp thất nghiệp không quá 2 ngày kể từ ngày nhận đủ hồ sơ theo quy định), cấp lại thẻ BHYT không thay đổi thông tin được thực hiện trong ngày. Với việc thực hiện giao dịch điện tử trong tất cả các khâu từ thu, nộp đến quản lý chi trả thì số lần thực hiện giao dịch điện tử giảm từ 12 lần/năm xuống còn 1 lần/năm. Cùng với việc cải cách thủ tục hành chính, năm 2020 và các năm tiếp theo các giải pháp bảo đảm điều kiện về cơ sở hạ tầng công nghệ thông tin; thực hiện giao dịch điện tử trên các lĩnh vực BHYT sẽ được triển khai bằng nhiều phương tiện trực tuyến khác nhau.

Kho dữ liệu (Data Warehouse) là hệ thống dữ liệu tập trung duy nhất, đầy đủ và nhất quán được tích hợp dữ liệu từ các nguồn khác nhau; được lưu trữ lịch sử lâu dài, nhằm mục đích thống kê, phân tích, hỗ trợ ra quyết định, tối ưu hóa các hoạt động của tổ chức mà không ảnh hưởng đến các phần mềm nghiệp vụ. Chương trình phần mềm tìm kiếm các thông tin liên quan đến BHYT thiết lập theo mô hình Kho dữ liệu Data Warehouse. Chương trình được xây dựng xuất phát từ yêu cầu thực tế: tổng hợp số liệu để xây dựng báo cáo theo yêu cầu của Lãnh đạo cấp trên, cũng như phục vụ công tác chỉ đạo, điều hành của Lãnh đạo cấp trên dựa trên thiết kế theo định hướng lấy đối tượng tham gia BHYT làm trung tâm, và ứng dụng thông tin chính xác.

3.2.1. Phân tích yêu cầu chức năng của ứng dụng

Trước đây, việc báo cáo, tổng hợp số liệu để xây dựng báo cáo theo yêu cầu của Lãnh đạo cấp trên cũng như phục vụ công tác chỉ đạo, điều hành của Lãnh đạo cấp trên được thực hiện trên các hệ thống phần mềm nghiệp vụ khác nhau. Mỗi hệ thống phần mềm chưa có sự liên thông, chuẩn hóa đôi khi có sự trùng lặp dẫn đến việc dữ liệu khó được đảm bảo tính toàn vẹn, chính xác và duy nhất và trong một số trường hợp còn thiếu. Điều đó dẫn đến những khó khăn trong việc tập hợp dữ liệu,

để lập các báo cáo, đưa ra các quyết định chiến lược. Thời gian thống kê số liệu cũng kéo dài do phải tổng hợp từ nhiều nguồn. Với công cụ chủ yếu để tổng hợp là file excel, các mẫu báo cáo không theo tiêu chuẩn, việc chỉnh sửa, bổ sung, thêm nội dung báo cáo khó có kết quả ngay trong khi rất nhiều yêu cầu cung cấp thông tin trong thời gian ngắn. Từ thực trạng của những hạn chế trên có thể thấy, nhu cầu về một kho dữ liệu tập trung và hệ thống các công cụ ứng dụng CNTT chuyên dụng phục vụ các công tác thống kê, báo cáo và hỗ trợ ra quyết định là cần thiết, cấp bách.

Việc triển khai hệ thống Data wearhouse nhằm xây dựng một kho dữ liệu tập trung về BHYT của ngành Công an là công cụ hỗ trợ ra quyết định với công nghệ hiện đại, tiên tiến và thông minh sẽ mang đến nhiều lợi ích hơn. Hệ thống này sẽ góp phần cải cách thủ tục hành chính, và ứng dụng công nghệ thông tin trong quản lý các hoạt động nghiệp vụ của BHYT ngành Công an.

Hệ thống Data warehouse sẽ được vận hành với hai cấu phần chính: xây dựng một kho dữ liệu của BHYT ngành Công an; và các công cụ hỗ trợ là các ứng dụng công nghệ thông tin để cung cấp thông tin dữ liệu mang tính chính xác, toàn vẹn và duy nhất cho các cấp quản lý của BHYT ngành Công an, cũng như mang lại nhiều cách thức khai thác dữ liệu chuyên nghiệp, linh hoạt, đa dạng hơn.

3.2.2. Phân tích hệ thống

Hiện tại, với 85 triệu người tham gia BHYT tại Việt Nam nên dữ liệu hộ gia đình tham gia BHYT với độ bao phủ gần 100% dân số nói chung, riêng với ngành Công an đến năm 2018 cán bộ, chiến sỹ tham gia BHYT đã đạt 100% quân số. Dữ liệu BHYT được xem là một trong các hệ dữ liệu quốc gia lớn và phức tạp hàng đầu với lượng thông tin đồ sộ, chứa đựng tất cả thông tin cơ bản của người tham gia BHYT từ lúc sinh ra đến khi mất đi, đi học, đi làm, lịch sử khám, chữa bệnh BHYT, các mối liên hệ gia đình để phục vụ việc giải quyết chế độ quyền lợi cho thân nhân cán bộ chiến sĩ...

Yếu tố quan trọng nhất trong việc xây dựng hệ thống Data warehouse là bắt đầu từ đâu? Phân tích dữ liệu như thế nào? Đây cũng là "nút thắt" cản trở khi quyết

định xây dựng hệ thống tương tự. Nguồn dữ liệu phân tán, được lưu trữ theo nhiều cách thức, tại nhiều đơn vị khác nhau, kể cả khi được mã hóa để lưu trữ điện tử cũng không có một chuẩn thống nhất là thực trạng chung đang diễn ra.

Từ năm 2015, sự "bùng nổ" ứng dụng công nghệ thông tin đã tạo ra sự đổi mới lớn. Hệ thống cơ sở dữ liệu được xây dựng theo hướng tập trung, sử dụng đường truyền Internet để chuyển dữ liệu thay vì chạy trên máy trạm như trước đây. Các dữ liệu cần được chuẩn hóa theo định dạng thống nhất để tạo nên hệ dữ liệu tập trung, được cập nhật theo từng ngày. Tuy vậy, để tạo lập được hệ thống kho dữ liệu tập trung của BHYT toàn ngành Công an đảm bảo độ chính xác, thì yêu cầu cao hơn là đưa tất cả các dữ liệu thuộc tất cả các lĩnh vực này vào một "kho" chung, và phải đảm bảo được truy xuất dễ dàng. Vì thế, dữ liệu đưa vào kho Data warehouse cần phải được chuẩn hóa, làm sạch để lưu trữ dữ liệu đã tập trung.

Mô hình Data warehouse không mới, nhưng đối với mỗi đơn vị vận hành sẽ có những yêu cầu khai phá dữ liệu ở mức độ phức tạp khác nhau. Nếu coi mỗi một yêu cầu phân tích dữ liệu là một đề bài, thì quá trình xây dựng hệ thống Data warehouse là hàng trăm đề bài khác nhau. Thực tế cho thấy, nhu cầu tổng hợp và phân tích dữ liệu của BHYT ngành Công an tập trung chủ yếu vào một số vấn đề như tổng số cán bộ, chiến sỹ tham gia BHYT và tổng số thân nhân cán bộ, chiến sỹ tham gia BHYT, khối thống kê, theo dõi xu hướng chi phí khám, chữa bệnh của các nhóm bệnh có rủi ro gian lận cao theo thời gian... nhằm phục vụ công tác quản lý, điều hành được hiệu quả hơn.

Chương trình phần mềm tìm kiếm các thông tin liên quan đến BHYT là phần mềm ứng dụng tập trung sẽ được cài đặt, lưu trữ tại Cục Chính sách- BCA và được triển khai tại các đơn vị. Hệ thống được thiết kế hỗ trợ việc thống kê, phân tích dữ liệu, thiết lập các báo cáo, hỗ trợ ra quyết định của Lãnh đạo Ngành. Trong đó, hệ thống Data warehouse sẽ được tích hợp với các hệ thống nguồn như sau:

1. Phần mềm Thu và quản lý thẻ BHYT.
2. Phần mềm Xét duyệt chính sách.
3. Phần mềm Giám định BHYT.

4. Phần mềm Kế toán tập trung.

5. Phần mềm Cấp mã số và quản lý thẻ BHYT.

6. Phần mềm Quản lý nhân sự.

7. Phần mềm Giao dịch điện tử.

8. Phần mềm Quản lý định danh và chia sẻ dữ liệu.

Từ những phân tích nêu trên, chương trình phần mềm tìm kiếm các thông tin liên quan đến BHYT cần được thiết kế dựa trên yêu cầu của 02 nhóm người sử dụng hệ thống Data warehouse.

3.2.2.1. Nhóm quản trị hệ thống Data warehouse

Là tập hợp những người sử dụng có chung đặc điểm, vai trò, tính chất và quyền hạn trong công tác quản trị, vận hành hệ thống. Trên hệ thống Data warehouse có các nhóm quản trị như sau:

- Quản trị ứng dụng cấp Bộ: Là cán bộ, chiến sỹ thuộc Cục Chính sách được Lãnh đạo cấp trên giao nhiệm vụ quản trị, vận hành ứng dụng tập trung.

- Quản trị ứng dụng cấp Tỉnh: Là cán bộ, chiến sỹ thuộc Phòng Chính sách Công an các tỉnh được Giám đốc Công an các tỉnh giao nhiệm vụ quản trị, vận hành ứng dụng trong địa bàn Công an tỉnh.

Nhiệm vụ của nhóm quản trị hệ thống Data warehouse:

1. Thực hiện cấp phát, thu hồi, thay đổi quyền truy cập hệ thống Data warehouse của người sử dụng.
2. Giám sát, hỗ trợ người sử dụng hệ thống Data warehouse trên địa bàn quản lý; phối hợp với các đơn vị liên quan xử lý các sự cố trong quá trình vận hành hệ thống Data warehouse.

3.2.2.2. Nhóm khai thác các chỉ tiêu báo cáo

Là tập hợp những người sử dụng có chung đặc điểm, vai trò, tính chất và quyền hạn thuộc đối tượng được lãnh đạo đơn vị giao nhiệm vụ sử dụng hệ thống Data warehouse để sử dụng hệ thống khai thác các chỉ tiêu, báo cáo.

Một người sử dụng có thể thuộc một hoặc nhiều nhóm người sử dụng tùy theo nhiệm vụ được phân công. Việc thay đổi nhóm người sử dụng trên hệ thống

Data warehouse được thực hiện trong trường hợp người sử dụng thay đổi vai trò, vị trí việc làm trong cơ quan hoặc luân chuyển giữa các cơ quan của ngành Công an.

Quy trình thay đổi nhóm người sử dụng được thực hiện theo quy trình thu hồi các quyền thuộc nhóm người sử dụng cũ sau đó cấp mới các quyền thuộc nhóm người sử dụng mới.

Nhiệm vụ của nhóm người sử dụng khai thác các chỉ tiêu, báo cáo:

1. Nhập bổ sung dữ liệu đối với các chỉ tiêu không có dữ liệu từ các hệ thống nguồn vào hệ thống Data warehouse.
2. Quản lý các chỉ tiêu, công thức các báo cáo được giao.
3. Tạo mới, điều chỉnh các chỉ tiêu, công thức báo cáo theo yêu cầu quản lý.
4. Vận hành, khai thác báo cáo theo yêu cầu quản lý đúng quy định.

Đặc biệt, do đặc thù ngành Công an dữ liệu của hệ thống Data werahouse ngoài tính chính xác và toàn vẹn cần phải được đảm bảo ở mức cao nhất về tính bảo mật. Trong mọi trường hợp, khi phát hiện nguy cơ mất an toàn từ phía người sử dụng, người phát hiện phải thông báo ngay cho Lãnh đạo đơn vị để chỉ đạo, kiểm tra và thông báo cho quản trị ứng dụng cấp Bộ hoặc quản trị ứng dụng cấp Tỉnh thu hồi quyền truy cập. Đây cũng là nguyên nhân chính thuật toán giấu thông tin mật trong cơ sở dữ liệu Data warehouse đã được xây dựng ở trên.

3.2.3. Giao diện của hệ thống

3.2.3.1. Giao diện để truy cập hệ thống

Quyền truy cập hệ thông Data warehouse là quyền truy cập gắn với địa chỉ hộp thư điện tử của người sử dụng được cấp quyền đăng nhập và truy cập vào hệ thống Data warehouse, thực hiện các nhiệm vụ được giao. Quyền truy cập này xác định danh tính, phạm vi tác nghiệp của người sử dụng trên hệ thống, bao gồm các yếu tố: Tên đăng nhập, mật khẩu đăng nhập và các quyền tương ứng với nhiệm vụ được giao.

- Hệ thống sử dụng dữ liệu của Phần mềm quản lý định danh và chia sẻ dữ liệu (IAM) để xác thực người sử dụng khi truy cập vào hệ thống Data warehouse.

Người sử dụng được cấp quyền truy cập hệ thống dùng tên và mật khẩu đăng nhập hệ thống IAM để đăng nhập, sử dụng hệ thống Data warehouse.

- Chỉ những tài khoản người sử dụng đã được thiết lập, phân quyền trên hệ thống ứng dụng mới có quyền đăng nhập sử dụng hệ thống Data warehouse.

3.2.3.2. Giao diện để truy cập báo cáo

Giao diện để truy cập báo cáo là công cụ dành cho người dùng khai thác, phân tích và phát triển các báo cáo phức tạp, nhiều trang, nhiều truy vấn với nhiều loại cơ sở dữ liệu. Người dùng có thể tạo bất kỳ báo cáo nào theo các yêu cầu quản lý như: tổng số thẻ bảo hiểm, tổng chi hàng tháng, chi một lần hay các hoạt động báo cáo cân đối kế toán…

- Các báo cáo tĩnh: Được thiết kế theo yêu cầu quản lý của ngành Công an, người dùng chỉ cần truy cập và thao tác nhập các tham số cần thiết để chạy báo cáo theo yêu cầu.

- Các chủ đề Phân tích động: Là các gói phân tích được thiết kế sẵn theo các chỉ tiêu và chiều phân tích theo nhiều chủ đề khác nhau. Phân tích động cho phép người dùng sử dụng giao diện để tạo các báo cáo, biểu đồ phân tích sâu, đánh giá và tìm hiểu các tác động dựa trên dữ liệu thực tế về công tác thu, chi, giám định BHYT, quản lý thẻ BHYT… để đưa ra các quyết định cụ thể và chính xác. Người dùng sẽ chủ động kéo thả dữ liệu cần khai thác dựa trên các chỉ tiêu, chiều được cung cấp trong từng chủ đề.

- Bảng thông tin tổng hợp (Dashboard): Là mẫu bảng chỉ tiêu được thiết kế gồm các bảng và biểu đồ số liệu giúp người dùng có cái nhìn từ tổng quát đến chi tiết về các chỉ tiêu khác nhau như thu, chi, giám định BHYT… một cách nhanh chóng để kịp thời đưa ra các quyết định.

3.2.4. Đánh giá hệ thống

Hệ thống kho dữ liệu phục vụ nhu cầu khai thác thông tin báo cáo, cùng các công cụ hỗ trợ phân tích động giúp cải thiện công tác báo cáo, thống kê BHYT của ngành Công an. Cụ thể:

- Với các báo cáo tổng hợp cơ bản từ các nguồn dữ liệu đã được xây dựng, người dùng nghiệp vụ hoàn toàn có thể khai thác dữ liệu mà không cần phụ thuộc quá nhiều vào đội ngũ cán bộ kỹ thuật công nghệ thông tin.

- Với các báo cáo tổng hợp phân tích phức tạp nhờ sự hỗ trợ của các công cụ phân tích báo cáo chuyên dụng, cao cấp, thời gian và công sức thực hiện cũng sẽ được giảm thiểu đáng kể.

Nhờ khả năng phân tích đa chiều, tra cứu thông tin lịch sử và nhiều tính năng cao cấp khác, người dùng nghiệp vụ có thể khai phá dữ liệu theo rất nhiều hướng tiếp cận, một cách linh hoạt nhất và từ đó sẽ có những cái nhìn sâu sát và trực quan, cũng như phát hiện ra nhiều khía cạnh thông tin (thực sự "hiểu" những thông tin mà dữ liệu bao hàm) mà các chức năng báo cáo thống kê hiện tại khó có thể mang lại. Với các tính năng này, hệ thống sẽ góp phần giải phóng người sử dụng, giúp tăng năng suất lao động, tiết kiệm thời gian để phục vụ tốt hơn các công tác chuyên môn thay vì phải thường xuyên can thiệp và tìm hiểu, thu thập dữ liệu một cách thủ công như trước đây.

3.2.4.1. Tính hiệu quả

Việc áp dụng Data warehouse sẽ giải quyết được vấn đề tích hợp, chia sẻ dữ liệu mà không phụ thuộc vào hệ điều hành và nền tảng công nghệ. Bên cạnh đó, việc triển khai kho dữ liệu là nền tảng tốt cho BHYT Công an, giúp lãnh đạo Y tế ngành Công an quản lý, điều hành và đưa ra quyết định một cách hiệu quả, chính xác và kịp thời.

Hiệu quả mà hệ thống tổng hợp và phân tích dữ liệu tập trung mang lại sẽ phát huy hơn nữa khi được tăng cường kết nối với các hệ thống báo cáo phục vụ cho công tác thống kê, tổng hợp, phân tích, đánh giá... Nhằm đưa ra các báo cáo phân tích các cấp, hỗ trợ công tác quản lý chặt chẽ, kịp thời, tổ chức thực hiện trở nên thiết thực và sâu sát hơn với các đối tượng tham gia BHYT của ngành Công an.

3.2.4.2. Các ưu và nhược điểm

Cần sớm triển khai xây dựng kho dữ liệu bởi vì nguồn dữ liệu của BHYT Công an ngày càng lớn, đa dạng và đã bộc lộ nhiều bất cập trong việc lưu trữ, quản

lý, chia sẻ, khai thác sử dụng, nếu không bắt tay ngay vào việc xây dựng kho dữ liệu, thì Y tế Công an không những không hạn chế được bất cập, mà còn làm bất cập tăng lên đến mức sẽ không thể kiểm soát được.

Tuy nhiên, quá trình xây kho dữ liệu sẽ gặp nhiều khó khăn và thách thức như: có rất nhiều loại dữ liệu trong hệ thống; logic nghiệp vụ phức tạp; nguồn nhân lực chưa đáp ứng được yêu cầu… Để giải quyết vấn đề này, học viên đề xuất giải pháp xây dựng kho dữ liệu đầu vào theo chủ đề từ đó từng bước xây dựng kho dữ liệu đầu vào tập trung.

1. Việc xây dựng kho dữ liệu theo từng giai đoạn, trước tiên sẽ xây dựng kho dữ liệu đầu vào theo hướng kho dữ liệu chủ đề (data mark), sau đó sẽ tích hợp các kho dữ liệu chủ đề thành kho dữ liệu tập trung. Xây dựng kho dữ liệu của BHYT ngành Công an sẽ dựa trên nền tảng công nghệ thông tin sẵn có của BHYT Công an nhân dân, đó là công nghệ khách/chủ. Hệ quản trị CSDL thích hợp nhất với công nghệ khách/chủ là Microsoft SQL server phiên bản 2018. Chuyển đổi dữ liệu sẽ sử dụng công cụ có sẵn là SQL Server Integration Services của Microsoft SQL server.

2. Vần đề an ninh, an toàn mạng và bảo mật dữ liệu (đã được đề cập ở chương II) là vấn đề quan trọng, cần được đầu tư đồng bộ với công nghệ khách/chủ, hệ quản trị CSDL.

3. Hơn nữa, xây dựng kho dữ liệu của BHYT ngành Công an là công việc rất lớn và mới của ngành, do đó, cần có sự hỗ trợ kỹ thuật của chuyên gia có kinh nghiệm xây dựng kho dữ liệu.

KẾT LUẬN

Trong toàn bộ Đề tài luận văn của mình, em đã giải quyết được 03 vấn đề cơ bản sau đây:

1. Nghiên cứu, tìm hiểu tổng quan về Data warehouse. Đây là vấn đề không mới nhưng hiện nay nó có nhiều ứng dụng trong thực tiễn, đặc biệt dối với an ninh-quốc phòng. Hiện tại, luận văn đã trình bày những nét cơ bản nhất của hệ thống Data warehouse với mục đích đưa vào ứng dụng trong Ngành Y tế - BCA.

2. Tìm hiểu và xây dựng một thuật toán giấu tin mật trong môi trường ảnh kỹ thuật số. Đây là một lĩnh vực về an toàn – bảo mật thông tin hiện nay đang phát triển mạnh trên thế giới [16]. Ở Việt Nam ta chỉ mới có ứng dụng trong Ngành Công an và chưa được phát triển nhiều. Thuật toán mà học viên xây dựng chủ yếu là sự mở rộng của thuật toán đã được công bố trong [7]. Cụ thể: học viên đã tìm hiểu và cải tiến thuật toán giấu tin 5 bit thành thuật toán giấu tin 6 bit, để tăng số lượng ký tự mã hóa (từ 31 ký tự thành 63 ký tự). Chương trình thể hiện thuật toán trên máy tính đã cho chạy thử nghiệm trên 21 mẫu (trang 35). Trong đó, có so sánh giữa 02 thuật toán và kết quả thuật toán cải tiến tốt hơn (trang 37).

3. Đề xuất phương pháp bảo vệ thông tin trong lĩnh vực y tế bằng hệ thống Data warehouse có bảo mật. Chương này chỉ mới phác thảo mục đích gợi ý cho bảo toàn thông tin trong ngành Y tế - BCA. Ứng dụng của hệ thống Data warehouse mới dừng ở việc lên ý tưởng xây dựng chương trình phần mềm tìm kiếm các thông tin liên quan đến Bảo hiểm y tế, chưa triển khai được trong thực tế.

Bên cạnh những kết quả đã đạt được, đề tài này cần được tiếp tục phát triển và hoàn thiện trong các năm tiếp theo. Do thời gian nghiên cứu có hạn và trình độ hiểu biết của bản thân còn nhiều hạn chế nên khóa luận của học viên không tránh khỏi những thiếu sót. Học viên rất mong nhận được sự góp ý quý báu của tất cả các thầy cô giáo để khóa luận của học viên được hoàn thiện hơn, góp phần đưa luận văn vào thực tiễn.

Học viên xin chân thành cảm ơn!

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. PGS.Nguyễn Kim Anh (2016). *Các hệ cơ sở dữ liệu tiên tiến*, Trường Đại học Bách Khoa Hà Nội.

[2]. TS.HồVăn Canh, TS.Nguyễn Viết Thế (2010). *Nhập môn phân tích thông tin có bảo mật,* Nhà xuất bản Thông tin và Truyền thông, trang 304 - 308.

[3]. PGS.TS.Nguyễn Đức Nghĩa (2014). *Thiết kế và Phân tích thuật toán*, Trường Đại học Bách Khoa Hà Nội.

[4]. PGS.TS.Thái Hồng Nhị, TS.Phạm Minh Việt (2004). *An toàn thông tin*, Nhà xuất bản Khoa học và kỹ thuật, 188 trang.

[5]. Nguyễn Văn Tảo (2007). *Một số thuật toán giấu tin và áp dụng giấu tin mật trong ảnh*, Tạp chí Khoa học và Công nghệ, số 4(44), Tập 2.

[6]. Hồ Thị Hương Thắm (2012). *Luận án Tiến sĩ: Nghiên cứu đề xuất phương pháp nhận dạng ảnh có chứa thông tin ẩn,* Đại học Công nghệ - Đại học Quốc Gia Hà Nội, 2012.

[7]. Lê Hải Triều (2019). *Luận án Tiến sĩ: Nghiên cứu phương pháp bảo mật thông tin giấu trong ảnh số,* Học viện Công nghệ Bưu chính Viễn thông – Hà Nội, 2019.

Tiếng Anh

[8]. Bruyn donckx, O.J.,J. Quisquater, and B. Macq, “Spatial Method for Copyright Labeling of Digital Images”, *In Non-linear Signal Processing Workshop, Thessloniki, Greece*, 1995, pp.456-459.

[9]. C-C-Raos (1968): “Linear Statisties and its applications”, NXB MOSCOW, 1968.

[10]. Chanfang yang, Xiangyang Leo, and Fenlin Liu: “Embedding Ratio Estimating for Each Bit plane of Image”, *Zenhgzhou Information Science and Technology Information Zhengzhou,* China 2015.

[11]. Cox, I., et al, “Seceere Spread Spectrum Watermarking for Multimedia”, *Technical report, NEC Research Institute*, 1995.

[12]. Depovere, G., T. Kalker, and J. – P. M. G. Linnartz, “Improved Watermark Detection Reliability Using Filtering Before Correlation”, *In Proceedings of the International Conference on Image Proceeding, vol. 1, IEEE Signal Proceeding Society, Chicago, ILLinois, USA*, oct. 1998.

[13]. Fisher, Y. (ed), Fractal Image Compression: Theory and Application, New York Springer-Verlag, 1995.

[14]. M. Warkentin, M. B. Schmidt, E. Bekkering (2008): “Steganography and Analysis”, *Premier Reference Source – Intellectual Property. Protection for Multimedia Information Technology, chapter XIX*, pp.374-380 (2008).

[15]. Puate, J., and F. Jordan, “Using Fractal Compression Scheme to Embed a Digital Signature in to an Image”, *In Proceedings of the SPIE 2015. Video Techniques and Software for Full-Service Networks*, 1996, pp.108-118.

[16]. R. Ibrahim and J.S. Kuan (2010): “Steganography Imageng Syotem (SIS), hiding Seeeret Message inside an Image”, *Proceedings of the world Congress on Engineering and Computer Science 2010 San Francisco*, USA, pp.144-148.

[17]. Rosziati Ibrahim and Jeoh Suk Kuan (2011): “Steganography Algorithm to Hide Seeret Message inside an Image”. *Computer Technology and Application 2* (2011).

[18]. Stefan Katzenbeisser, Fabien A. P. Petitcolas: (2000): “Information Hiding Techniques for Steganography and Digital watermarking”, *Artech House Boston. London*.

[19]. T. Jahnke, J. Seitz (2008): “An Introduction in digital watermarking Applications, Porneiples and problems”, *in: H. Nemati (Ed), Security and Ethics: Concepts, Methodologies, Tools and Applications. NewYork: Information Science Reference*, pp.554-569.

PHỤ LỤC

A/ Hàm phi-ơle Ø(.) được định nghĩa như sau:

Với n là số tự nhiên. Khi đó

Ø(1) = 1

Ø(n) = |{1 ≤ i < n : gcd(i, n) = 1}|

Trong đó, gcd(i, n) là ước số chung lớn nhất của i và n (i và n nguyên tố cùng nhau thì gcd(i, n) = 1)

Các tính chất của Ø(n):

Tính chất 1. $Ø(n) = n. \prod_{p/n} (1 - \frac{1}{p})$ (p là ước số nguyên tố của n)

Tính chất 2. Cho 2 số m và n nguyên tố cùng nhau tức là gcd(m, n) = 1.

Khi đó, Ø(m.n) = Ø(m) . Ø(n)

Tính chất 3. Nếu n = p là số nguyên tố thì Ø(p) = p – 1

Tính chất 4. $Ø(n) > \frac{n}{6\ln\ln n}$

Tính chất 5. Nếu $n = p^k$, trong đó p là số nguyên tố và k ≥ 1

thì $Ø(n) = Ø(p^k) = p^{k-1} (p-1)$

Tính chất 6. Nếu gcd(a, n) = 1 với a là số nguyên dương và n ≥ 2

thì $a^{Ø(n)} \equiv 1$ modn

Tính chất 7. Nếu p, q là 2 số nguyên tố khác nhau

thì Ø(p.q) = (p-1) (q-1)

B/ Chương trình nguồn của thuật toán giấu thông tin mật

- **HÀM RANDOM KHÓA**

```
public static string RandomKey(int length)
        {
            const string chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
            return new string(Enumerable.Repeat(chars, length)
              .Select(s => s[random.Next(s.Length)]).ToArray());
        }
```

- **MÃ HÓA THÔNG ĐIỆP**

```
   private static string[] arrCumTu = new string[] { "tr/loi" };
```

```
        private static Dictionary<string, string> dicMaHoa = new Dictionary<string,
string>()
    {
        { " ", "000000"},
        { "a", "010000"},
        { "b", "001000"},
        { "c", "000100"},
        { "d", "000010"},
        { "e", "000001"},
        { "f", "010100"},
        { "g", "001010"},
        { "h", "000101"},
        { "i", "010110"},
        { "j", "001011"},
        { "k", "010001"},
        { "l", "011100"},
        { "m", "001110"},
        { "n", "000111"},
        { "o", "010111"},
        { "p", "011111"},
        { "q", "011011"},
        { "r", "011001"},
        { "s", "011000"},
        { "t", "001100"},
        { "u", "000110"},
        { "v", "000011"},
        { "w", "010101"},
        { "x", "011110"},
        { "y", "001111"},
        { "z", "011101"},
        { "/", "011010"},
        { "0", "100000"},
        { "1", "110000"},
        { "2", "111000"},
        { "3", "111100"},
        { "4", "111110"},
        { "5", "111111"},
        { "6", "101000"},
        { "7", "101100"},
```

```
    { "8", "101110"},
    { "9", "101111"},
    { "A", "100100"},
    { "B", "100110"},
    { "C", "100111"},
    { "D", "100010"},
    { "E", "100011"},
    { "F", "100001"},
    { "G", "101010"},
    { "H", "101011"},
    { "I", "100101"},
    { "J", "110001"},
    { "K", "110010"},
    { "L", "110111"},
    { "M", "110011"},
    { "N", "101001"},
    { "O", "110100"},
    { "P", "111101"},
    { "Q", "111011"},
    { "R", "101101"},
    { "T", "111001"},
    { "U", "110101"},
    { "V", "111010"},
    { "W", "010011"},
    { "X", "001001"},
    { "Y", "010010"},
    { "Z", "001101"},
};

    public static string MaHoaThongDiep(string text)
    {
        // text = text.Replace(" ", "*");
        text = text.Trim();
        text = text.ToLower();
        string tu;
        int len = arrCumTu.Length;
        string[] arrt = text.Split(' ');
        string textTemp = "";
        for (int i = 0; i < arrt.Length; i++)
```

```
            {
                if (arrt[i] == "ong")
                {
                    arrt[i] = dicMaHoa["ong"];
                }
                if (i > 0) textTemp += " ";
                textTemp += arrt[i];

            }

            for (int i = 0; i < len; i++)
            {
                tu = arrCumTu[i];
                textTemp = textTemp.Replace(tu, dicMaHoa[tu]);
            }

            foreach (char c in textTemp)
            {
                if (dicMaHoa.ContainsKey(c.ToString()))
                {
                    textTemp = textTemp.Replace(c.ToString(), dicMaHoa[c.ToString()]
+ " ");
                }
            }
            textTemp = textTemp.Trim();
            return textTemp = textTemp.Replace("  ", " ");

        }
```

- HÀM MÃ HÓA VÀ GIẢI MÃ

```
private static byte[] _salt =
Encoding.ASCII.GetBytes("jasdh7834y8hfeur73rsharks214");

        /// <summary>
        /// Encrypt the given string using AES.  The string can be decrypted using
        /// DecryptStringAES().  The sharedSecret parameters must match.
        /// </summary>
        /// <param name="plainText">The text to encrypt.</param>
        /// <param name="sharedSecret">A password used to generate a key for encryption.</param>
```

```
        public static string EncryptStringAES(string plainText, string sharedSecret)
        {
            if (string.IsNullOrEmpty(plainText))
                throw new ArgumentNullException("plainText");
            if (string.IsNullOrEmpty(sharedSecret))
                throw new ArgumentNullException("sharedSecret");

            string outStr = null;           //Encrypted string to return
            RijndaelManaged aesAlg = null;  // RijndaelManaged object used to encrypt the data.

            try
            {
                // generate the key from the shared secret and the salt
                Rfc2898DeriveBytes key = new Rfc2898DeriveBytes(sharedSecret,
_salt);

                // Create a RijndaelManaged object
                aesAlg = new RijndaelManaged();
                aesAlg.Key = key.GetBytes(aesAlg.KeySize / 8);

                // Create a decryptor to perform the stream transform.
                ICryptoTransform encryptor = aesAlg.CreateEncryptor(aesAlg.Key,
aesAlg.IV);

                // Create the streams used for encryption.
                using (MemoryStream msEncrypt = new MemoryStream())
                {
                    // prepend the IV
                    msEncrypt.Write(BitConverter.GetBytes(aesAlg.IV.Length), 0,
sizeof(int));
                    msEncrypt.Write(aesAlg.IV, 0, aesAlg.IV.Length);
                    using (CryptoStream csEncrypt = new CryptoStream(msEncrypt,
encryptor, CryptoStreamMode.Write))
                    {
                        using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
                        {
                            //Write all data to the stream.
                            swEncrypt.Write(plainText);
                        }
```

```
                }
                outStr = Convert.ToBase64String(msEncrypt.ToArray());
            }
        }
        finally
        {
            // Clear the RijndaelManaged object.
            if (aesAlg != null)
                aesAlg.Clear();
        }

        // Return the encrypted bytes from the memory stream.
        return outStr;
    }

    /// <summary>
    /// Decrypt the given string.  Assumes the string was encrypted using
    /// EncryptStringAES(), using an identical sharedSecret.
    /// </summary>
    /// <param name="cipherText">The text to decrypt.</param>
    /// <param name="sharedSecret">A password used to generate a key for decryption.</param>
    public static string DecryptStringAES(string cipherText, string
sharedSecret)
    {
        if (string.IsNullOrEmpty(cipherText))
            throw new ArgumentNullException("cipherText");
        if (string.IsNullOrEmpty(sharedSecret))
            throw new ArgumentNullException("sharedSecret");

        // Declare the RijndaelManaged object
        // used to decrypt the data.
        RijndaelManaged aesAlg = null;

        // Declare the string used to hold
        // the decrypted text.
        string plaintext = null;

        try
        {
```

```
                // generate the key from the shared secret and the salt
                Rfc2898DeriveBytes key = new Rfc2898DeriveBytes(sharedSecret,
_salt);

                // Create the streams used for decryption.
                byte[] bytes = Convert.FromBase64String(cipherText);
                using (MemoryStream msDecrypt = new MemoryStream(bytes))
                {
                    // Create a RijndaelManaged object
                    // with the specified key and IV.
                    aesAlg = new RijndaelManaged();
                    aesAlg.Key = key.GetBytes(aesAlg.KeySize / 8);
                    // Get the initialization vector from the encrypted stream
                    aesAlg.IV = ReadByteArray(msDecrypt);
                    // Create a decrytor to perform the stream transform.
                    ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key,
aesAlg.IV);
                    using (CryptoStream csDecrypt = new CryptoStream(msDecrypt,
decryptor, CryptoStreamMode.Read))
                    {
                        using (StreamReader srDecrypt = new StreamReader(csDecrypt))

                            // Read the decrypted bytes from the decrypting stream
                            // and place them in a string.
                            plaintext = srDecrypt.ReadToEnd();
                    }
                }
            }
            finally
            {
                // Clear the RijndaelManaged object.
                if (aesAlg != null)
                    aesAlg.Clear();
            }

            return plaintext;
        }
```

- **HÀM GIẤU TIN VÀ LẤY TIN TỪ ẢNH**

```
public enum State
      {
          Hiding,
          Filling_With_Zeros
      };

      public static Bitmap embedText(string text, Bitmap bmp)
      {
          // initially, we'll be hiding characters in the image
          State state = State.Hiding;

          // holds the index of the character that is being hidden
          int charIndex = 0;

          // holds the value of the character converted to integer
          int charValue = 0;

          // holds the index of the color element (R or G or B) that is currently being processed
          long pixelElementIndex = 0;

          // holds the number of trailing zeros that have been added when finishing the process
          int zeros = 0;
          int charcount = 0;
          // hold pixel elements
          int R = 0, G = 0, B = 0;
          //ghi thông điệp từ vị trí x
          int y= bmp.Height;
          int x = bmp.Width;
          // pass through the rows
          for (int i = 0; i < y; i++)
          {
              // pass through each row
              for (int j = 0; j < x; j++)
              {
                  // holds the pixel that is currently being processed
                  Color pixel = bmp.GetPixel(j, i);
```

```
// now, clear the least significant bit (LSB) from each pixel element
R = pixel.R - pixel.R % 2;
G = pixel.G - pixel.G % 2;
B = pixel.B - pixel.B % 2;

// for each pixel, pass through its elements (RGB)
for (int n = 0; n < 3; n++)
{
    // check if new 8 bits has been processed
    if (pixelElementIndex % 8 == 0)
   {
        // check if the whole process has finished
        // we can say that it's finished when 8 zeros are added
        if (state == State.Filling_With_Zeros && zeros == 8)
        {
            // apply the last pixel on the image
            // even if only a part of its elements have been affected
            if ((pixelElementIndex - 1) % 3 < 2)
            {
                bmp.SetPixel(j, i, Color.FromArgb(R, G, B));
            }

            // return the bitmap with the text hidden in
            return bmp;
        }

        // check if all characters has been hidden
        if (charIndex >= text.Length)
        {
            // start adding zeros to mark the end of the text
            state = State.Filling_With_Zeros;
        }
        else
        {
            // move to the next character and process again
            charValue = text[charIndex++];
        }
   }
```

```
// check which pixel element has the turn to hide a bit in its LSB
switch (pixelElementIndex % 3)
{
    case 0:
        {
            if (state == State.Hiding)
            {
                // the rightmost bit in the character will be (charValue % 2)
                // to put this value instead of the LSB of the pixel element
                //just add it to it
                // recall that the LSB of the pixel element had been cleared
                // before this operation
                R += charValue % 2;

                // removes the added rightmost bit of the character
                // such that next time we can reach the next one
                charValue /= 2;
                charcount++;
            }
        } break;
    case 1:
        {
            if (state == State.Hiding)
            {
                G += charValue % 2;

                charValue /= 2;
                charcount++;
            }
        } break;
    case 2:
        {
            if (state == State.Hiding)
            {
                B += charValue % 2;

                charValue /= 2;
                charcount++;
            }
```

```
                            bmp.SetPixel(j, i, Color.FromArgb(R, G, B));
                        } break;
                }

                pixelElementIndex++;

                if (state == State.Filling_With_Zeros)
                {
                    // increment the value of zeros until it is 8
                    zeros++;
                }
            }
        }
    }

    return bmp;
}

public static string extractText(Bitmap bmp)
{
    int colorUnitIndex = 0;
    int charValue = 0;

    // holds the text that will be extracted from the image
    string extractedText = String.Empty;
    //ghi thông điệp từ vị trí y
    int y = bmp.Height;
    int x = bmp.Width;
    // pass through the rows
    for (int i = 0; i < y; i++)
    {
        // pass through each row
        for (int j = 0; j <x; j++)
        {
            Color pixel = bmp.GetPixel(j, i);

            // for each pixel, pass through its elements (RGB)
            for (int n = 0; n < 3; n++)
```

```
{
    switch (colorUnitIndex % 3)
    {
        case 0:
            {
            // get the LSB from the pixel element (will be pixel.R % 2)
            // then add one bit to the right of the current character
            // this can be done by (charValue = charValue * 2)
            // replace the added bit (which value is by default 0) with
            // the LSB of the pixel element, simply by addition
                charValue = charValue * 2 + pixel.R % 2;
            } break;
        case 1:
            {
                charValue = charValue * 2 + pixel.G % 2;
            } break;
        case 2:
            {
                charValue = charValue * 2 + pixel.B % 2;
            } break;
    }

    colorUnitIndex++;

    // if 8 bits has been added, then add the current character to the result text
    if (colorUnitIndex % 8 == 0)
    {
        // reverse? of course, since each time the process happens on the right (for simplicity)
        charValue = reverseBits(charValue);

        // can only be 0 if it is the stop character (the 8 zeros)
        if (charValue == 0)
        {
            return extractedText;
        }

        // convert the character value from int to char
        char c = (char)charValue;
```

```
                    // add the current character to the result text
                    extractedText += c.ToString();
                }
            }
        }
    }

    return extractedText;
}

public static int reverseBits(int n)
{
    int result = 0;

    for (int i = 0; i < 8; i++)
    {
        result = result * 2 + n % 2;

        n /= 2;
    }

    return result;
}
```