LỜI CẢM ƠN

Trong quá trình nghiên cứu và hoàn thiện Đồ án tốt nghiệp, tôi đã nhận được rất nhiều sự giúp đỡ và đóng góp quý báu.

Đầu tiên, tôi xin bày tỏ lòng cảm ơn sâu sắc tới thầy giáo hướng dẫn là Thạc sĩ Vũ Tuấn Minh - Giáo viên Khoa Công nghệ thông tin đã luôn ủng hộ, động viên, tận tình giúp đỡ và hỗ trợ điều kiện tốt nhất cho tôi trong suốt quá trình nghiên cứu và hoàn thiện Đồ án tốt nghiệp.

Tôi xin gửi lời cám ơn chân thành tới các các chuyên gia của EmbeddedSW Company, các thầy cô giáo trong Khoa Công nghệ thông tin, các đồng chí học viên lớp D3A đã góp ý, giúp đỡ tôi hoàn thành đề tài này.

Cuối cùng, tôi xin bày tỏ lòng cảm ơn đến gia đình, anh em, bạn bè, các đồng chí đồng đội đã động viên và cổ vũ tôi trong suốt thời gian nghiên cứu.

Mặc dù tôi đã có nhiều cố gắng, nhưng do trình độ còn hạn chế, thời gian thực hiện chưa nhiều, nên trong báo cáo không thể tránh khỏi sai sót. Tôi rất mong nhận được sự góp ý của tất cả mọi người để hoàn thiện báo cáo này.

Xin trân trọng cảm ơn tất cả!

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Nghĩa Tiếng Anh	Nghĩa Tiếng Việt
AES	Advanced Encryption Standard	Tiêu chuẩn mã hóa cấp cao
CBC	Cipher Block Chaining	Mã hóa chuỗi khối
CRC	Cyclic Redundancy Check	Kiểm tra dư thừa tuần hoàn
CRYPTREC	Cryptography Research and Evaluation Committees	Ủy ban nghiên cứu và đánh giá mật mã
CSPRNG	Cryptographically Secure Pseudorandom Number Generator	Bộ tạo số giả ngẫu nhiên mật mã
CTR	Counter	Bộ đếm
DES	Data Encryption Standard	Tiêu chuẩn mã hoá dữ liệu
IV	Initialization Vector	Véc tơ khởi tạo
KDF	Key Derivation Function	Hàm dẫn xuất khóa
LSB	Least Significant Bit	Bít ít quan trọng nhất
NESSIE	New European Schemes for Signatures, Integrity and Encryption	Các chương trình Châu Âu mới về chữ ký, tính toàn vẹn và mã hóa
NIST	National Institute of Standards and Technology	Viện Tiêu chuẩn và Công nghệ quốc gia Hoa Kỳ
SHA	Secure Hash Algorithm	Thuật toán băm bảo mật

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG

DANH MỤC CÁC KÝ HIỆU TOÁN HỌC

Ký hiệu	**Ý nghĩa**
$\oplus$	Phép toán XOR
E_k	Mã hóa E với khóa K
Rand-i ()	Hàm Random i
C_i	Khối bản mã thứ i
P_i	Khối bản rõ thứ i
O_i	Khối mã hóa thứ i
I_i	Bộ đếm thứ i

MỞ ĐẦU

1. Lý do chọn đề tài

Cùng với sự phát triển của khoa học kỹ thuật, thì việc bảo mật thông tin ngày càng thể hiện vai trò quan trọng của nó. Các tổ chức, tập thể, cá nhân ngày càng quan tâm đến vấn đề bảo mật các thông tin, dữ liệu của mình. Nhiều khi chỉ một sơ suất nhỏ làm lộ lọt thông tin có thể dẫn đến hậu quả rất nghiêm trọng.

Hơn thế nữa, ngày nay việc truyền gửi dữ liệu càng trở nên phổ biến và cần thiết. Vì vậy vấn đề bảo mật cho các dữ liệu để đảm bảo an toàn cho quá trình gửi và nhận là không thể thiếu. Không ít các thuật toán mã hóa đã được sử dụng để đảm bảo an toàn cho dữ liệu. Nhưng chúng vẫn có một số rủi ro nhất định, các thuật toán mã hóa cũng dần dần bị con người tìm ra cách giải mã.

Chính vì thế, mô hình bảo mật 4 lớp được sáng tạo và phát triển. Mô hình này thể hiện sự vượt trội về tính bảo mật so với các mô hình bảo mật thông thường. Khi sử dụng mô hình này dữ liệu không chỉ được mã hóa mà còn được ẩn dưới lớp vỏ bọc là các dữ liệu đa phương tiện bình thường nhờ các kỹ thuật che dấu thông tin, nhằm tránh gây sự chú ý cho những người khác.

Do đó với nhu cầu sử dụng, tiềm năng ứng dụng của mô hình bảo mật 4 lớp, được sự đồng ý hướng dẫn của thầy giáo Thạc sĩ Vũ Tuấn Minh, tôi đã chọn thực hiện đề tài *"Nghiên cứu mô hình bảo mật 4 lớp, áp dụng vào giấu tin trong dữ liệu đa phương tiện"* để làm Đồ án tốt nghiệp.

Việc nghiên cứu tìm hiểu về vấn đề này là cơ hội để tôi trau dồi kiến thức về lĩnh vực an toàn thông tin cụ thể là về bảo mật thông tin, mã hóa dữ liệu và dấu tin, nâng cao khả năng tư duy, hiểu biết về các thuật toán mã hóa, các kỹ thuật dấu tin, vận dụng các kiến thức, kỹ năng đã học để giải quyết một vấn đề cụ thể.

Kết quả của đề tài trước hết là một nguồn tài liệu phục vụ cho sinh viên CNTT, sau đó đóng góp một phần nhỏ bé vào việc hỗ trợ các công tác Công an, thực hiện đúng tinh thần phát huy nội lực công nghệ thông tin của Ngành. Tri thức học tập và nghiên cứu được từ đề tài giúp nâng cao trình độ, năng lực bản thân, đáp ứng yêu cầu nhiệm vụ trong công tác Công an của đơn vị sau khi tốt nghiệp ra trường.

2. Các công trình nghiên cứu có liên quan

Dự án bán mã nguồn mở OpenPuff là công trình duy nhất ứng dụng mô hình bảo mật 4 lớp. Còn ở thời điểm hiện tại, cả trong và ngoài nước chưa có công trình nào khác nghiên cứu chuyên sâu về mô hình bảo mật 4 lớp này.

Tuy nhiên có một số công trình có một phần nội dung nghiên cứu liên quan đến mô hình bảo mật 4 lớp, cụ thể như:

1. Nguyễn Thanh Cường (2009), "Giấu tin trong ảnh và ứng dụng trong an toàn bảo mật thông tin", *Khóa luận tốt nghiệp*, *Trường Đại Học Công Nghệ*, *Đại Học Quốc Gia Hà Nội*: Đây là một luận văn có nội dung tập trung vào việc trình bày một số khái niệm liên quan đến Steganography và nghiện cứu về kỹ thuật Steganography được áp dụng trong ảnh.

2. Yao Lu (2014), "Investigating Steganography in Audio Stream for Network Forensic Investigations: Detection and Extraction", *Auckland*, *New Zealand*: Một luận văn có nội dung chính là nghiên cứu Steganography trong các luồng âm thanh cho lĩnh vực điều tra số trên mạng.

3. Cosimo Oliboni (2011), "OpenPuff Help: Steganography & Watermarking", *Italy*: Một tài liệu tham khảo chính thống của dự án bán mã nguồn mở OpenPuff. Có nội dung chính là giới thiệu về mô hình bảo mật 4 lớp, phần mềm OpenPuff và cách sử dụng.

4. Michael Chesbro (2014), "OpenPuff: Steganography & Watermarking Tool": Một bài giảng về giới thiệu Steganography và phần mềm OpenPuff.

5. Cosimo Oliboni (2011), "Multiobfuscator: Cryptography & Obfuscation", *Italy*: Một tài liệu tham khảo của dự án mã nguồn mở Multiobfuscator, tiền thân của OpenPuff. Có nội dung chính là giới thiệu về bộ lõi Multiobfuscator, thứ sau này được sử dụng để phát triển mô hình bảo mật 4 lớp và phần mềm OpenPuff.

3. Mục đích nghiên cứu

Đồ án tốt nghiệp thực hiện nhiệm vụ "Nghiên cứu mô hình bảo mật 4 lớp, áp dụng vào giấu tin trong dữ liệu đa phương tiện" nhằm mục đích tạo tiền đề lý thuyết cơ sở cho việc tìm hiểu về mô hình bảo mật 4 lớp. Đồng thời nghiên cứu việc áp dụng mô hình này vào việc giấu tin trong dữ liệu đa phương tiện nhờ sử dụng phần mềm OpenPuff.

4. Nhiệm vụ nghiên cứu

Sinh viên thực hiện đề tài: "*Nghiên cứu mô hình bảo mật 4 lớp, áp dụng vào giấu tin trong dữ liệu đa phương tiện*", với nhiệm vụ cụ thể như sau:

1. Trình bày, báo cáo tổng quan về mô hình bảo mật 4 lớp, áp dụng vào giấu tin trong dữ liệu đa phương tiện.

2. Áp dụng được mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện bằng phần mềm OpenPuff.

5. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là:

1. Mô hình bảo mật 4 lớp.

2. Các thuật toán mã hóa, dấu tin liên quan đến mô hình bảo mật 4 lớp.

3. Phần mềm bán mã nguồn mở OpenPuff.

6. Phương pháp nghiên cứu

Phương pháp nghiên cứu bao gồm:

1. Phương pháp phân tích và tổng hợp lý thuyết: Phân tích, nghiên cứu các tư liệu, tài liệu, lý luận từ nhiều nguồn, phân tích chúng thành từng bộ phận để tìm hiểu sâu sắc về mô hình bảo mật 4 lớp. Tổng hợp, liên kết từng mặt, từng bộ phận thông tin đã được phân tích tạo ra một hệ thống lý thuyết mới đầy đủ và sâu sắc để trình bày vào báo cáo.

2. Phương pháp tham khảo tài liệu: Tài liệu về lĩnh vực Steganograph, tài liệu về bảo mật thông tin, thuật toán trên Internet, báo cáo khoa học, tài liệu hội thảo.

3. Phương pháp tham khảo ý kiến chuyên gia: Tham khảo ý kiến của Lãnh đạo Khoa, các thầy cô giáo có nhiều kinh nghiệm và các Cán bộ Công an địa phương.

4. Phương pháp thực nghiệm: Tiến hành cài đặt, nghiên cứu sử dụng phần mềm OpenPuff ẩn giấu dữ liệu.

7. Phạm vi nghiên cứu

Trong khuôn khổ phục vụ cho Đồ án tốt nghiệp hệ Đại học chính quy, đề tài được định hướng có phạm vi nghiên cứu tập trung vào việc nghiên cứu

về mô hình bảo mật 4 lớp, áp dụng mô hình bảo mật 4 lớp vào việc giấu tin trong dữ liệu đa phương tiện bằng cách sử dụng phần mềm bán mã nguồn mở OpenPuff.

8. Các đóng góp của ĐATN

Kết quả nghiên cứu của đề tài sẽ là tài liệu về mô hình bảo mật 4 lớp trong lĩnh vực Steganography, cách thức áp dụng mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện. Sản phẩm cũng là một tài liệu tham khảo có tính chuyên môn dành cho sinh viên chuyên ngành Công nghệ thông tin nói chung và sinh viên của Trường Đại học Kỹ thuật - Hậu cần CAND nói riêng.

9. Bố cục của ĐATN

Bố cục của ĐATN gồm có 3 chương:

- Chương 1: Tổng quan về Steganography.

- Chương 2: Mô hình bảo mật 4 lớp và thuật toán liên quan.

- Chương 3: Phần mềm thử nghiệm áp dụng mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện.

Chương 1
TỔNG QUAN VỀ STEGANOGRAPHY

Chương 1 tập trung giới thiệu một cách tổng quan nhất về định nghĩa, lịch sử, các khái niệm cơ bản trong lĩnh vực Steganography. Trình bày các phương pháp, kỹ thuật Steganography trong an toàn thông tin. Nhằm mục đích tạo cho chúng ta một cái nhìn chung nhất về lĩnh vực này.

1.1. Giới thiệu về Steganography

Steganography (kỹ thuật giấu tin) là kỹ thuật che giấu một tập tin, thông điệp, hình ảnh hoặc video trong các tập tin, thông điệp, hình ảnh hoặc video khác. Thuật ngữ Steganography được hình thành nhờ sự kết hợp của những từ Hi Lạp là: Steganos, có nghĩa là "bao phủ, che dấu hoặc bảo vệ" và Graphein, có nghĩa là "viết".

Các ghi nhận đầu tiên về thuật ngữ này do Johannes Trithemius sử dụng trong cuốn sách Steganographia của ông vào năm 1499, một luận thuyết về mật mã và giấu tin được ngụy trang như một cuốn sách ma thuật. Các thông diệp ẩn được ẩn hoặc trở thành một phần của một thứ hoàn toàn khác như: Hình ảnh, bài viết, danh sách mua sắm hoặc một số dạng văn bản che giấu khác [7].

Yêu cầu quan trọng nhất của bất kỳ hệ thống Steganography nào là con người hoặc máy tính không thể phân biệt một cách dễ dàng giữa các đối tượng bình thường và đối tượng có chứa dữ liệu bí mật [3].

Steganography bao gồm cả việc che giấu thông tin trong các tập tin máy tính. Trong lĩnh vực giấu tin kỹ thuật số, các thông tin liên lạc điện có thể bao gồm giấu tin mã hóa bên trong tầng vận chuyển, chẳng hạn như một tập tin tài liệu, tập tin ảnh, phần mềm hoặc giao thức. Các tập tin đa phương tiện rất phù hợp cho việc truyền giấu tin bởi vì chúng có kích thước tập tin lớn [7].

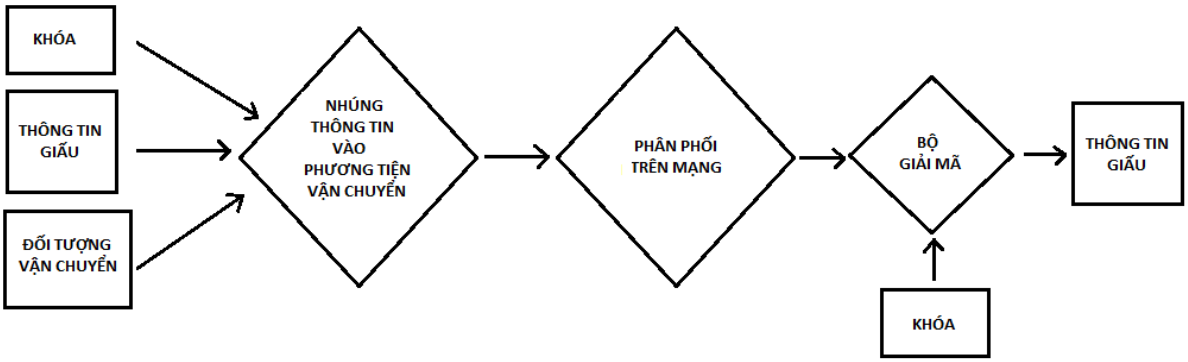
Steganography gồm 2 thành phần chính:

- Thuật toán giấu tin.

- Bộ giải mã thông tin.

Thuật toán giấu tin được dùng để giấu thông tin vào một đối tượng vận chuyển bằng cách sử dụng một khóa bí mật được dùng chung bởi người mã hóa và người giải mã, việc giải mã thông tin chỉ có thể thực hiện được khi có khoá. Bộ giải mã thực hiện giải mã trên đối tượng vận chuyển đã chứa dữ liệu để trả lại thông điệp ẩn trong nó.

Quá trình Steganography cơ bản được mô tả như sau:



Hình 1.1. Lược đồ chung cho quá trình Steganography.

Trong đó:

- Đối tượng vận chuyển bao gồm các đối tượng được dùng làm môi trường để giấu tin như: Text, audio, video, ảnh,…

- Giấu thông tin là mục đích của người sử dụng, thông tin giấu là một lượng thông tin mang một ý nghĩa nào đó như ảnh, logo, đoạn văn bản… tùy thuộc vào mục đích của người sử dụng.

- Thông tin sẽ được giấu vào trong đối tượng vận chuyển nhờ một bộ nhúng, bộ nhúng là những phần mềm, triển khai các thuật toán để giấu tin và được thực hiện với một khóa bí mật giống như các hệ mã cổ điển.

- Sau khi giấu tin ta thu được đối tượng vận chuyển chứa thông tin bí mật và được phân phối sử dụng trên mạng.

- Sau khi nhận được đối tượng vận chuyển 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 khóa của quá trình nhúng.

- Kết quả thu được gồm đối tượng vận chuyển ban đầu và thông tin đã giấu. Cuối cùng thông tin giấu sẽ được xử lý kiểm định để xác nhận nguồn gốc của thông tin [1].

Mục đích chính của Steganography là giấu đi những thông điệp bí mật khỏi việc kiểm duyệt, gián điệp hay kẻ trộm. Những thông điệp tưởng chừng như vô hại và nhàm chán cũng có thể thu hút sự chú ý.

Những thông điệp đã được mã hóa có thể sẽ không bị giải mã và phát hiện bởi những thế lực thù địch, nhưng việc sử dụng mã hóa có thể làm kẻ địch tin rằng thông tin có chứa thông tin nhạy cảm hoặc không hợp pháp dù nó có

hay không. Ở một số nơi, việc sử dụng mức độ bảo mật cao có thể bị hạn chế hoặc bị cấm. Steganography cho phép thông điệp đã mã hóa được ẩn đi và tránh khỏi những sự tò mò của người khác.

Trong khi Cryptography (mật mã học) chỉ là việc bảo vệ các nội dung của một thông điệp đơn thuần, Steganography còn liên quan với việc che giấu sự thật trong quá trình truyền gửi một thông điệp bí mật, cũng như che giấu nội dung của thông điệp.

Ưu điểm của Steganography là những thông điệp bí mật được chủ ý không để thu hút sự chú ý đến bản thân nó. Rõ ràng ta có thể nhìn thấy các thông điệp đã được mã hóa, tránh được sự tò mò, quan tâm của những người khác.

Steganography làm rất tốt trong việc giấu đi 1 lượng nhỏ dữ liệu bí mật. Hạn chế lớn nhất của Steganography là các tập tin vận chuyển phải lớn hơn đáng kể so với những dữ liệu đã được ẩn. Ví dụ, ta không thể ẩn nội dung của 1 cuốn sách lớn trong 1 tệp hình ảnh nhỏ [5].

Tóm lại, Steganography là kỹ thuật giấu một truyền thông bên trong một truyền thông khác, với mục đích là che giấu những thông điệp bí mật mà không làm ảnh hưởng đến thông điệp chung. Kỹ thuật này giúp người dùng bình thường khó có thể phát hiện và giải mã các thông điệp bí mật được ẩn chứa trong các thông điệp khác, giúp tăng cường tính an toàn thông tin trong quá trình truyền tải dữ liệu [1].

1.2. Lịch sử về Steganography

Những ghi nhận về việc sử dụng Steganography đầu tiên là từ năm 440 trước công nguyên, khi Herodotus, một nhà sử học Hy Lạp, đã ghi chép lại hai ví dụ trong lịch sử của mình:

- Ghi chép thứ nhất về Histiaeus, một bạo chúa của Miletus vào cuối thế kỷ thứ 6 trước công nguyên, đã gửi một thông điệp đến chư hầu của ông, Aristagoras, bằng cách cạo đầu đầy tớ đáng tin cậy nhất của mình, đánh dấu thông điệp lên da đầu của hắn, chờ tóc mọc lại và gửi hắn cho Aristagoras với sự hướng dẫn là hãy cạo đầu mình khi gặp Aristagoras.

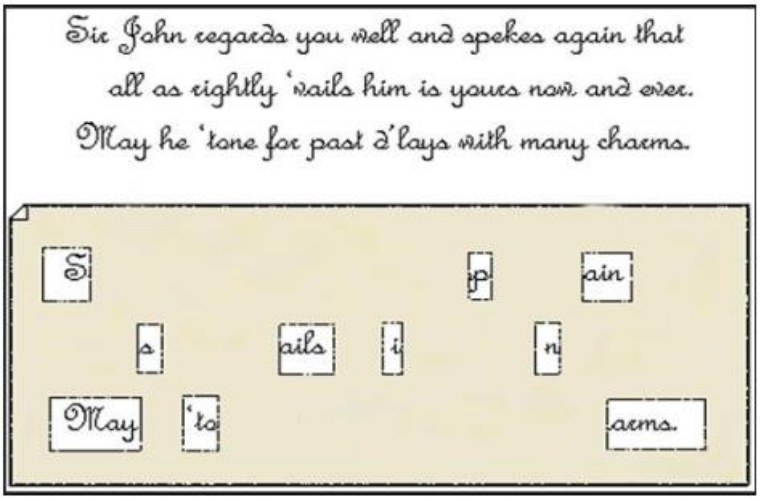
- Ghi chép thứ 2, Demaratus, một vị vua của Sparta, đã gửi một lời cảnh báo về một cuộc tấn công sắp tới của Hy Lạp bằng cách viết nó trực tiếp vào lớp bồi bằng gỗ của một bảng sáp trước khi được phủ sáp ong lên nó. Bảng sáp

được sử dụng phổ biến sau đó như tái sử dụng bề mặt để viết bản thể hoặc dùng để viết các văn bản ký hiệu [7].

Thuật ngữ Steganography lần đầu tiên được sử dụng bởi Johnnes Trithemius (1462-1516) trong bộ ba tác phẩm Polygraphia và trong Steganographia. Các nhà học giả khi đó nghi ngờ rằng cuốn sách đã chứa một mật mã và đã cố gắng để giải mã bí ẩn đó, thứ mà đã không thể được giải quyết cho đến năm 1996.

Dường như các phương pháp Steganography sớm được sử dụng để ẩn thông điệp trong văn bản, nó cũng được gọi là ngôn ngữ Steganography hoặc là Acrostics.

Một dự án về ngôn ngữ Steganography nổi tiếng có tên là Cardan's Grilled, nó ban đầu được hình thành ở Trung Quốc và được xây dựng lại bởi Gerolamo Cardano, một học giả người Ý. Nó đơn giản chỉ là những vùng che đậy văn bản trên bức thư chứa thông điệp bí mật.



Hình 1.2. Mẫu thư chứa Steganography của Carda's Grille.

Một mẫu phương pháp nâng cao từ Cardan's Grille, nó được gọi là Turning Grill đã được sử dụng Thế chiến thứ nhất.

Bề ngoài, Turning Grill trông giống như một tấm lưới bình thường. Nhưng để sử dụng Turning Grille, bộ mã hóa được thực hiện bằng cách viết dãy thứ nhất trong bức thư, sau đó xoay tấm lưới 90 độ và viết dòng thứ 2 của bức thư, và cứ tiếp tục như vậy, xoay tấm lưới sau mỗi dòng một. Sau đó thông điệp bí mật được tạo thành.

Một trường hợp khác của Steganography trong Thế chiến thứ nhất là viết thông điệp bí mật sử dụng mực không màu. Các gián điệp đã sử dụng sữa, giấm và nước hoa quả để viết thông điệp trên một tờ giấy đen và nó không thể nhìn thấy bởi mắt người. Người nhận có thể nhìn thấy thông điệp bằng cách làm nóng tờ giấy.

Trong Thế chiến thứ hai, kỹ thuật Steganography đã được phát triển và ứng dụng. Nazis đã tạo ra phương pháp gọi là Microdot. Một Microdot là một bức ảnh nhỏ của một chu kỳ đánh máy. Nó có thể tái sản xuất theo tiêu chuẩn kích thước trang đánh máy với độ rõ nét tuyệt vời, cực kỳ khó để phát hiện. Sau đó, người Đức sử dụng Microdot để truyền số lượng lớn dữ liệu in trong chiến tranh.

Vào năm 1980, Thủy vân vô hình đã được sử dụng bởi cựu thủ tướng Anh, Margaret Thatcher. Sau khi một số tài liệu nội bộ đã bị rò rỉ, Thatcher đã yêu cầu tất cả hệ thống xử lý tài liệu mã hóa danh tính của họ trên các khoảng trống của tài liệu. Điều này cho phép các bộ trưởng không trung thành nhanh chóng bị phát hiện.

Hiện nay, các hình thức của Steganography kỹ thuật số có tầm quan trọng đặc biệt. Có một số công nghệ kỹ thuật số mà cộng đồng quan tâm như: tập tin văn bản, ảnh, video, audio.

Đa số các tổ chức sử dụng công nghệ Steganography để bảo vệ sở hữu trí tuệ. Các tổ chức khác thì sử dụng công nghệ Steganography có liên quan với Digital Watermark (thủy vân số) cho việc bảo vệ tài sản. Sử dụng nhiều kỹ thuật, các tập tin ảnh, nhạc, phim, có thể in dấu với các Digital Watermark [3]. Về khái niệm Digital Watermark sẽ được trình bày cụ thể ở phần sau.

Steganography đang phát triển nhanh chóng như một công nghệ bảo mật, có vai trò quan trọng trong nhiều khía cạnh của lĩnh vực công nghệ thông tin nói chung cũng như lĩnh vực an toàn thông tin nói riêng. Các phần tiếp theo sẽ trình bày cụ thể về các nội dung của công nghệ Steganography này.

1.3. Các khái niệm cơ bản trong Steganography

Để hiểu rõ hơn về Steganography cũng như dễ dàng tìm hiểu hiểu các nội dung trong các phần tới, phần này sẽ đề cập đến một số khái niệm cần chú ý trong lĩnh vực Steganography:

1. Giấu tin: Là quá trình ẩn một dữ liệu vào trong một môi trường dữ liệu khác. Dữ liệu trước khi giấu có thể được nén và mã hoá bằng nhiều cách.

Trong các ứng dụng đòi hỏi độ bảo mật cao, việc giấu dữ liệu chính là một phương pháp bảo mật thông tin hiệu quả. Việc giấu dữ liệu được ứng dụng trong nhiều lĩnh vực khác nhau như bảo vệ bản quyền, ngăn ngừa sao chép trái phép, truyền thông bí mật v.v..

2. Giấu tin trong dữ liệu đa phương tiện: Là một phần của khái niệm "giấu tin" với việc sử dụng dữ liệu đa phương tiện làm phương tiện vận chuyển. Giấu thông tin trong dữ liệu đa phương tiện có nhiều ứng dụng trong thực tế như trong việc xác định quyền sở hữu, chống xuyên tạc thông tin và chuyển giao dữ liệu một cách an toàn.

3. Secret Messages (thông điệp bí mật): Là các thông tin, dữ liệu quan trọng mà chúng ta cần che dấu, đảm bảo bí mật và cần được bảo vệ trong quá trình truyền tải.

4. Carrier (đối tượng vận chuyển): Là tính hiệu, luồng hoặc tệp dữ liệu mà trong đó dữ liệu ẩn được che dấu bằng những cách sửa đổi tin vi. Ví dụ như: Tập tin âm thanh, hình ảnh, tài liệu và các tập tin thi hành. Trong thực tế, Carrier nên bắt chước và làm việc giống như các Carrier ban đầu chưa được sửa đổi, và nên có vẻ "lành tính" với bất cứ ai muốn kiểm tra nó.

Một số thuộc tính dẫn đến việc nghi ngờ một tệp tin đang chứa dữ liệu ẩn:

+ Nếu dữ liệu ẩn có kích thước lớn so với nội dung của nhà cung cấp, như kích thước 1 megabyte trong một tập tin tài liệu trống.

+ Việc sử dụng các định dạng đã lỗi thời hoặc phần mở rộng được hỗ trợ kém.

Một yêu cầu mật mã là các Carrier phải là bản gốc, không phải bản sao của một thứ gì đã được công khai, ví dụ như đã được tải lên Internet. Vì mã nguồn dữ liệu công khai có thể được đối chiếu với phiên bản có chứa một thông điệp ẩn đã được nhúng vào.

Một yêu cầu nữa là thông điệp được nhúng không thay đổi số liệu thống kê của một Carrier (hoặc các chỉ số khác) vì sự hiện diện của một thông điệp có thể bị phát hiện [8].

5. Carrier Engine: Là lõi của bất cứ một công cụ Steganography nào. Các định dạng tệp khác nhau được sửa đổi theo nhiều cách khác nhau để dữ liệu ẩn được ẩn bên trong chúng. Các thuật toán xử lý bao gồm:

+ Injection (tiêm thuốc)

+ Generation (thế hệ)

+ Dữ liệu phụ (Ancillary data) và thay thế siêu dữ liệu (Metadata Substitution).

+ LSB (Least Significant Bit) hoặc thay thế thích ứng (adaptive substitution).

+ Thao tác không gian tần số (Frequency space manipulation) [8].

6. Carrier Chain (chuỗi đối tượng vận chuyển): Dữ liệu ẩn có thể được phân chia giữa một tập hợp các tập tin, tạo ra một Carrier Chain, trong đó có các tính chất mà tất cả các đối tượng vận chuyển phải có là: Tính khả dụng, tính toàn vẹn và phải được xử lý theo thứ tự chính xác để lấy các dữ liệu ẩn.

Tính năng bảo mật bổ sung này thường đạt được bằng cách:

+ Sử dụng Vector khởi tạo khác nhau cho mỗi đối tượng vận chuyển và lưu trữ nó bên trong các đối tượng vận chuyển đã được xử lý, điều này có nghĩa là: $\text{CryptedIV}_n = \text{Crypt}(\text{IV}_n, \text{CryptedIV}_{n-1})$.

+ Sử dụng một thuật toán mật mã khác cho mỗi đối tượng vận chuyển và lựa chọn nó bằng một thuật toán chuỗi thứ tự phụ thuộc [8].

7. Modified Carrier (ModCarrier): Là các đối tượng vận chuyển sau khi đã được sửa đổi, cũng có nghĩa là chúng đã được nhúng các thông điệp bí mật.

8. Embedded (nhúng): Cách viết khác Embedding, Imbedded hoặc Imbedding, là quá trình nhúng các thông điệp bí mật vào trong một dữ liệu môi trường nào đó, ví dụ như các tập tin vận chuyển.

9. Obfuscation (sự xáo trộn): Là cố ý che dấu về ý nghĩa có dụng ý của truyền thông, thường bằng cách làm cho thông điệp trở nên dễ nhầm lẫn, mơ hồ hoặc khó hiểu [9].

10. Noise (nhiễu): Là sự xuất hiện những đặc điểm lạ của thông điệp bí mật cần được nhúng so với lúc ban đầu. Việc này nhằm mục đích là thông điệp bí mật trở nên khó hiểu để được bảo mật cao hơn.

11. Digital Watermark (thủy vân số): Là một loại đánh dấu bí mật được dấu trong một tín hiệu chứa nhiễu như dữ liệu âm thanh, hình ảnh hoặc video. Nó thường được sử dụng để xác định quyền sở hữu bản quyền của tính hiệu đó.

Digital Watermark là quá trình giấu thông tin kỹ thuật số trong một tín hiệu Carrier, nhưng không cần có một mối quan hệ với các tính hiệu Carrier đó.

Digital Watermark có thể được sử dụng để xác minh tính xác thực hoặc tính toàn vẹn của tín hiệu Carrier hoặc để hiển thị danh tính của chủ sở hữu. Nó được sử dụng rộng rãi để theo dõi vi phạm bản quyền và để xác thực tiền giấy.

Giống như các Physical Watermark, Digital Watermark thường chỉ nhận biết được trong điều kiện nhất định, ví dụ như sau khi sử dụng một số thuật toán.

Nếu một Digital Watermark làm thay đổi tín hiệu Carrier theo cách mà nó trở nên dễ nhận thấy, nó có thể được coi là kém hiệu quả tùy thuộc vào mục đích sử dụng của nó.

Các Watermark truyền thống có thể được áp dụng cho các dữ liệu đa phương tiện có thể nhìn thấy, giống như hình ảnh hoặc video, trong khi ở Digital Watermarking, tín hiệu có thể là âm thanh, hình ảnh, video, văn bản hoặc mô hình 3D.

Một tín hiệu có thể mang nhiều Watermark khác nhau cùng một lúc. Không giống như siêu dữ liệu được thêm vào tín hiệu Carrier, một Digital Watermark không thay đổi kích thước của tín hiệu Carrier.

Các thuộc tính cần thiết của một Digital Watermark phụ thuộc vào các trường hợp sử dụng nơi mà nó được áp dụng. Để đánh dấu các tệp phương tiện với thông tin bản quyền, một Digital Watermark phải tương đối mạnh mẽ chống lại những sửa đổi có thể được áp dụng cho tín hiệu Carrier. Thay vào đó, nếu đảm bảo tính toàn vẹn, một Watermark dễ gãy vỡ sẽ được áp dụng.

Cả Steganography và Digital Watermark đều sử dụng các kỹ thuật Steganographic để nhúng dữ liệu bí mật trong tín hiệu nhiễu. Tuy nhiên, trong khi Steganography nhằm mục đích không nhận thấy đối với các giác quan của con người, Digital Watermarking cố gắng kiểm soát tính mạnh mẽ như là ưu tiên hàng đầu. Digital Watermark là một công cụ bảo mật thụ động. Nó chỉ đánh dấu dữ liệu chứ không làm suy giảm nó hoặc kiểm soát truy cập vào dữ liệu [10].

12. Tính không nhìn thấy: Là một trong ba yêu cầu của bất kỳ một hệ giấu tin nào.Tính không nhìn thấy là tính chất vô hình của thông tin nhúng trong phương tiện nhúng.

13. Tính mạnh mẽ: Là yêu cầu thứ hai của một hệ giấu tin. Tính mạnh mẽ là nói đến khả năng chịu được các thao tác biến đổi nào đó trên phương tiện nhúng và các cuộc tấn công có chủ đích.

14. Khả năng nhúng: Là yêu cầu thứ ba của một hệ giấu tin. Khả năng nhúng chính là số lượng thông tin được nhúng trong phương tiện chứa [1].

1.4. Steganography trong an toàn thông tin

Phần này sẽ giới thiệu về các phương pháp Steganography được sử dụng trong lĩnh vực an toàn thông tin. Đồng thời phân loại và làm nổi bật vai trò của Steganography trong an toàn thông tin. Giới thiệu về kỹ thuật giấu tin LSB.

1.4.1. Phân loại Steganography trong an toàn thông tin

Trong an toàn thông tin, có rất nhiều phương pháp Steganography mà hầu hết chúng ta đều quen thuộc: Từ mực không màu đến những bức ảnh Microdot,... Với sự phát triển của máy tính và Internet, chúng ta có nhiều cách khác để giấu thông tin hơn trong lĩnh vực an toàn thông tin, chẳng hạn như:

- Những kênh bí mật. Ví dụ, Loki và một số công cụ từ chối dịch vụ phân tán sử dụng Giao thức ICMP - Internet Control Message Protocol, là kênh truyền thông giữa những kẻ xấu và một hệ thống bị xâm nhập.

- Ẩn văn bản trong các trang Web.

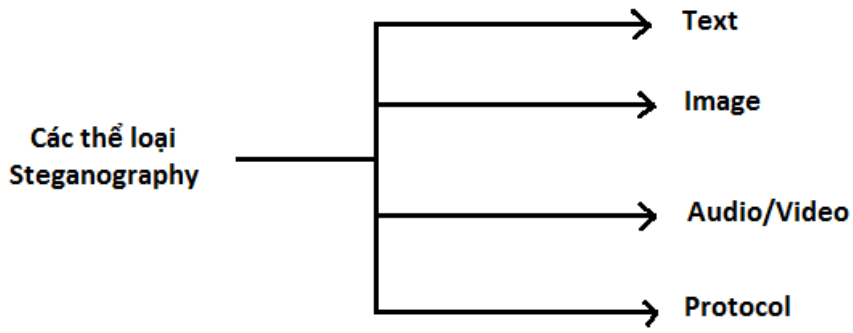
- Ẩn các tập tin trong các “Plain Sight”. Ví dụ ẩn tập tin với tên là các tập tin âm thanh quan trọng trong thư mục C:\Winnt\System32.

- Mật khẩu Null. Ví dụ, sử dụng chữ cái đầu tiên của mỗi từ để tạo thành một thông điệp ẩn trong một văn bản vô hại.

Tuy nhiên, Steganography ngày nay phức tạp hơn nhiều so với các ví dụ trên, cho phép người dùng ẩn nhiều lượng thông tin trong các tệp hình ảnh và âm thanh. Các hình thức Steganography này thường được sử dụng kết hợp với mật mã để thông tin được bảo vệ kép. Thông tin được mã hóa và ẩn để trước hết kẻ tấn công phải tìm ra thông điệp, một nhiệm vụ thường gặp nhiều khó khăn và sau đó là phải giải mã nó.

Hiện nay, có rất nhiều loại phương pháp Steganography và chúng hỗ trợ hầu hết các định dạng tập tin số.

Hình sau đây cho thấy các dạng tệp tin khác nhau có thể sử dụng cho kỹ thuật Steganography:



Hình 1.3. Các thể loại của Steganography.

Nhìn chung, chỉ có 3 phương pháp trong ẩn dấu một thông điệp kỹ thuật số trong lĩnh vực công nghệ thông tin là:

- Injection (tiêm thuốc)

- Substitution (thay thế)

- Genaration of new files (tạo ra các tập tin mới)

Injection có nghĩa là trực tiếp nhúng thông điệp bí mật vào trong vật chủ môi trường. Vấn đề là nó thường tạo ra các tập tin vật chủ lớn và sau đó làm nó trở nên dễ dàng bị phát hiện.

Substitution có nghĩa là thay thế dữ liệu bình thường bằng dữ liệu bí mật. Nó sẽ giữ nguyên kích thước của tập tin vật chủ nhưng có thể làm mất một số dữ liệu của tập tin vật chủ nguyên bản.

Genaration of new files, như cái tên của nó, là tạo ra một vỏ bọc duy nhất để ẩn dữ liệu bí mật đi.

Với 3 phương pháp trên, chúng ta có thể chia thành 6 công nghệ sau:

- Subtitution System Techniques.

- Transform Domain Techniques.

- Spread Spectrum Techniques.

- Statistical Method Techniques.

- Distortion Techinques.

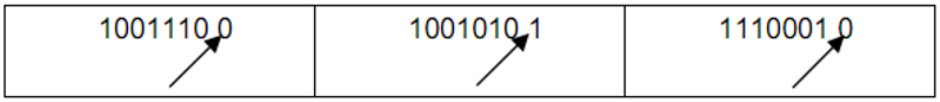
- Cover Generation Techniques [3].

1.4.2. Giới thiệu kỹ thuật LSB

LSB (Least Signification Bit) hay còn gọi là bít ít quan trọng nhất, là một kỹ thuật giấu tin được sử dụng phổ biến trong các tập tin hình ảnh. Ý tưởng của thuật toán này là tiến hành giấu tin vào các vị trí bít ít quan trọng nhất của mỗi phần tử trọng bản màu.

Đây là phương pháp giấu tin đơn giản nhất, thông điệp dưới dạng nhị phân sẽ được giấu (nhúng) vào các bit LSB - là bit có ảnh hưởng ít nhất tới việc quyết định tới màu sắc của mỗi điểm ảnh. Vì vậy khi ta thay đổi bit ít quan trọng của một điểm ảnh thì màu sắc của mỗi điểm ảnh mới sẽ tương đối gần với điểm ảnh cũ. Ví dụ đối với ảnh 16 bit thì 15 bit là biểu diễn 3 màu RGB của điểm ảnh còn bit cuối cùng không dùng đến thì ta sẽ tách bit này ra ở mỗi điểm ảnh để giấu tin…

Ví dụ: Tách bit cuối cùng trong 8 bit biểu diễn mỗi điểm ảnh của ảnh 256 màu.



Hình 1.4. Biểu diễn bit ít quan trọng nhất trong kỹ thuật LSB

Trong phép tách này ta coi bit cuối cùng là bit ít quan trọng nhất, thay đổi giá trị của bit này thì sẽ thay đổi giá trị của điểm ảnh lên hoặc xuống đúng một đơn vị, với sự thay đổi nhỏ đó ta hi vọng là cấp độ màu của điểm ảnh sẽ không bị thay đổi nhiều. [2]

Ví dụ thực hiện giấu chữ cái "A" có mã ASCII là 65, được biểu diễn dưới dạng bit là 01000001 vào trong 8 byte của tập tin gốc:

Bảng 1.1. Ví dụ giấu chữ cái A vào trong 8 byte đầu của tập tin gốc.

8 byte ban đầu	Byte cần giấu (A)	8 byte sau khi giấu
01001001	0	01001000
11010111	1	11010111
11001100	0	11001100
10110101	0	10110100
00100100	0	00100100
00100101	0	00100100

00100000	0	00100000
00001010	1	00001011

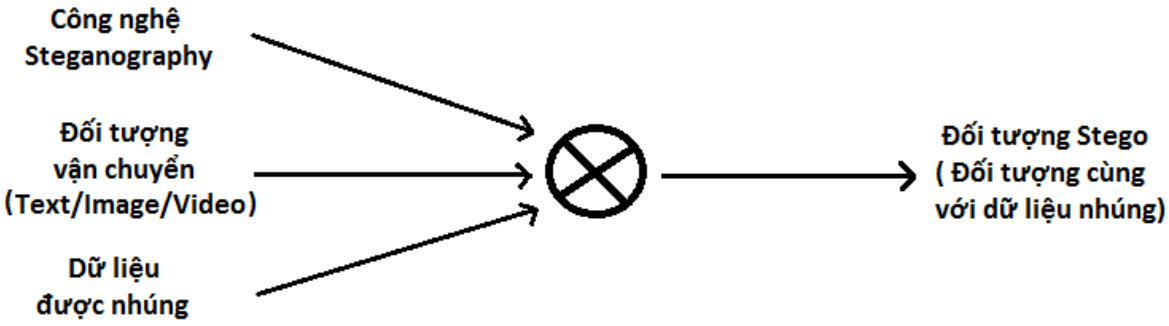
1.4.3. Vai trò của Steganography trong an toàn thông tin

Steganography có vai trò rất quan trong trong lĩnh vực an toàn thông tin. Nó được sử dụng trong việc che dấu và bảo mật các thông điệp bí mật, quan trọng được truyền gửi trên các môi trường truyền thông như môi trường mạng.

Trong an toàn thông tin, Steganography có liên quan mật thiết với Cryptography. Cryptography là quá trình mã hóa các thông điệp để người khác không thể dễ dàng hiểu được bởi một người không được ủy quyền. Mặt khác, Steganography sẽ ẩn thông điệp để người khác không thể biết về sự tồn tại của thông điệp đó.

Nếu một người nào đó xem các đối tượng chứa thông tin ẩn sẽ không thể nghĩ rằng có thông tin được ẩn trong đó. Do đó, họ sẽ không cố gắng để giải mã thông điệp.

Quá trình giấu tin trong Steganography được biểu diễn như hình dưới đây.



Hình 1.5. Mô hình chung của quá trình giấu tin trong Steganography.

Trong đó:

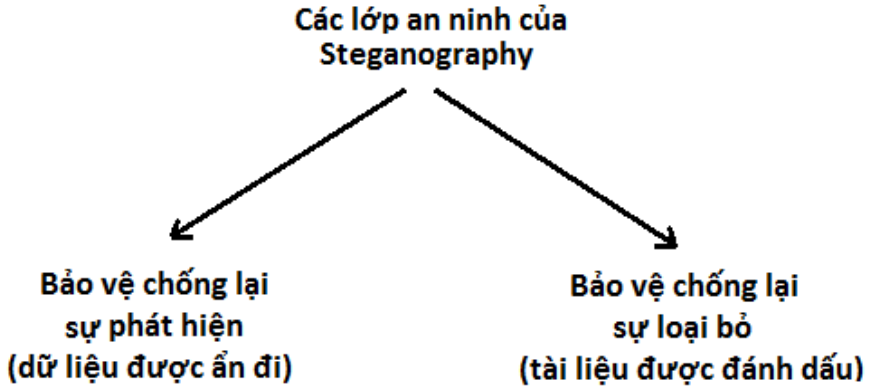
- Các thông điệp bí mật có thể được nhúng vào các đối tượng vận chuyển bởi bộ mã hóa Stegosystem cùng việc sử dụng khóa mật khẩu.

- Một thông điệp bí mật có thể là văn bản thô, văn bản mật mã, một bức ảnh hoặc bất cứ thứ gì khác có thể biểu diễn được dưới dạng một luồng bit.

- Một khi đối tượng vận chuyển được nhúng thông tin vào thì nó được gọi là đối tượng Stego.

- Sau quá trình nhúng, đối tượng Stego được gửi đến người nhận. Người nhận sử dụng bộ giải mã thích hợp cùng với khóa mật khẩu để tìm ra bản gốc của thông điệp người gửi muốn truyền đạt.

Một kỹ thuật Steganography cần thỏa mãn 2 yêu cầu:



Hình 1.6. Lớp an ninh của Steganography.

- Bảo vệ chống lại sự phát hiện (Protection Against Detection) rất cần thiết nếu muốn đảm bảo rằng thông điệp được nhúng không được phát hiện bởi bên thứ ba trái phép.

- Bảo vệ chống lại sự loại bỏ (Protection Against Removal), có nghĩa là cố gắng để ngăn chặn việc loại bỏ các dữ liệu ẩn, không làm hỏng hoặc làm suy giảm chất lượng của nó.

Mặc dù có rất nhiều lợi ích và những công nghệ hiện đại trong Steganography, nhưng trong lĩnh vực truyền thông, việc đảm bảo an toàn thông tin là nhiệm vụ quan trọng nhất.

Với sự tiến bộ của công nghệ và việc sử dụng rộng rãi của World Wide Web cho truyền thông đã làm tăng thêm các thách thức trong an ninh, an toàn về thông tin.

Các công nghệ tiên tiến đã làm cho các trao đổi thông tin qua mạng an toàn hơn. Nhưng khi mà các công nghệ này không đảm bảo sự tin cậy cho việc truyền thông bí mật trên một khoảng cách dài thì chúng ta cần phải tạo ra một cơ chế an ninh bổ sung. Không thể không kể đến vai trò của hai lĩnh vực Cryptography và Steganography. Chúng giúp bảo mật thông tin ngay cả khi chúng ta bị lộ lọt trong đường truyền.

Trong khi Cryptography giữ bí mật thông tin bằng cách thay đổi ý nghĩa và thông tin xuất hiện của thông điệp, thì Steganography sử dụng cơ chế bảo

mật bằng cách dấu các thông điệp bí mật vào các thông điệp thông thường khác. Các nội dụng được nhúng làm thay đổi kích cỡ không đáng kể cho các thông điệp bị nhúng.

Chính vì các lý do trên, Steganography ngày càng thể hiện sự quan trong của mình trong lĩnh vực an toàn thông tin.

Phần này đã trình bày tương đối rõ nét về vai trò của Steganography trong lĩnh vực an toàn thông tin làm tiền đề cơ sở để chúng ta tiếp tục nghiên cứu sâu về mô hình của nó trong các chương tiếp theo, cụ thể là nghiên cứu mô hình bảo mật 4 lớp ở chương 2.

1.5. Kết luận chương 1

Chương này đã hoàn thành việc trình bày một cách tổng quan nhất về lĩnh vực Steganography gồm các nội dung sau:

- Giới thiệu về Steganography: Trình bày định nghĩa, cách thức làm việc, lược đồ chung, mục đích sử dụng, ưu điểm, hạn chế của Steganography.

- Lịch sử về Steganography: Quá trình hình thành và phát triển của Steganography.

- Các khái niệm cơ bản trong Steganography: Trình bày các khái niệm thường gặp nhất trong Steganography như: *Giấu tin, giấu tin trong dữ liệu đa phương tiện, Secret Messages, Carrier, Carrier Engine, Carrier Chain, Modified Carrier, Embedded, Obfuscation, Noise, Digital Watermark, tính không nhìn thấy, tính mạnh mẽ, khả năng nhúng.*

- Steganography trong an toàn thông tin: Giới thiệu về Steganography trong lĩnh vực an toàn thông tin, mô hình hoạt động chung, các lớp an ninh của Steganography. Trình bày các thể loại, các phương pháp kỹ thuật số của Steganography, tầm quan trọng của nó trong lĩnh vực này.

Việc giới thiệu tổng quan về Steganography và các khái niệm liên quan ở chương này nhằm mục đích tạo nền tảng kiến thức cơ sở phục vụ cho việc tìm hiểu các thuật toán, các lớp trong mô hình bảo mật 4 lớp sẽ trình bày ở Chương 2.

Chương 2
MÔ HÌNH BẢO MẬT 4 LỚP VÀ THUẬT TOÁN LIÊN QUAN

Nội dung của chương 2 sẽ trình bày lý thuyết về giới thiệu tổng quan, kiến trúc mô hình và các thuật toán cơ bản trong mô hình bảo mật 4 lớp. Làm rõ quá trình mã hóa của từng lớp trong mô hình bảo mật này, đặc biệt là đi sâu hơn vào 2 lớp đầu Layer 1 và 2, còn 2 lớp Layer 3 và 4 thì chỉ nghiên cứu ở mức độ hiểu biết do vấn đề độc quyền của mô hình bảo mật 4 lớp. Ngoài ra, chương này còn đưa ra một số nhận xét về ưu nhược điểm của mô hình, giúp chúng có thể dễ dàng đánh giá và so sánh mô hình này với các mô hình bảo mật khác.

2.1. Tổng quan về mô hình bảo mật 4 lớp

2.1.1. Giới thiệu về mô hình bảo mật 4 lớp

Mô hình bảo mật 4 lớp (4 Layers Security) là một mô hình bảo mật thông tin do EmbeddedSW Company, một công ty chuyên sản xuất các hệ thống nhúng đã sáng tạo và phát triển nhằm mục đích mã hóa và ẩn giấu các thông điệp bí mật trong các tập tin vận chuyển. Mô hình này đã được chính EmbeddedSW Company ứng dụng trong dự án bán mã nguồn mở OpenPuff của mình. Mô hình bảo mật 4 lớp được phát triển dựa trên hệ thống mã nguồn mở LibObfuscate.

Hệ thống mã nguồn mở LibObfuscate là một hệ thống thực hiện Multi-Cryptography, hay còn gọi là đa mật mã, một loại mật mã nâng cao của mã hóa xác suất với 16 thuật toán mã hóa khối hiện đại mã nguồn mở, được chọn trong số: AES-Process, NESSIE-Process và CRYPTREC-Process.

LibObfuscate còn sử dụng thuật toán Cypher-Block-Chaining (CBC) để thực hiện các thuật toán mã hóa dựa trên khối, cho phép chúng hoạt động như các thuật toán dựa trên luồng.

Mô hình bảo mật 4 lớp gồm: Modern Multi-Cryptography, CSPRNG Based Scrambling, CSPRNG Based Whitening, Adaptive Non-Linear Encoding. Trong đó:

- Hai lớp đầu tiên là Modern Multi-Cryptography và CSPRNG Based Scrambling được sáng tạo và phát triển dựa trên hệ thống mã nguồn mở LibObfuscate.

- Hai lớp còn lại là CSPRNG Based Whitening và Adaptive Non-Linear Encoding được EmbeddedSW Company phát triển như một công nghệ độc quyền của họ, nên ở phạm vi đồ án này chỉ dừng lại ở việc tìm hiểu chung về 2 lớp này chứ không thể đi sâu vào việc phân tích chi tiết.

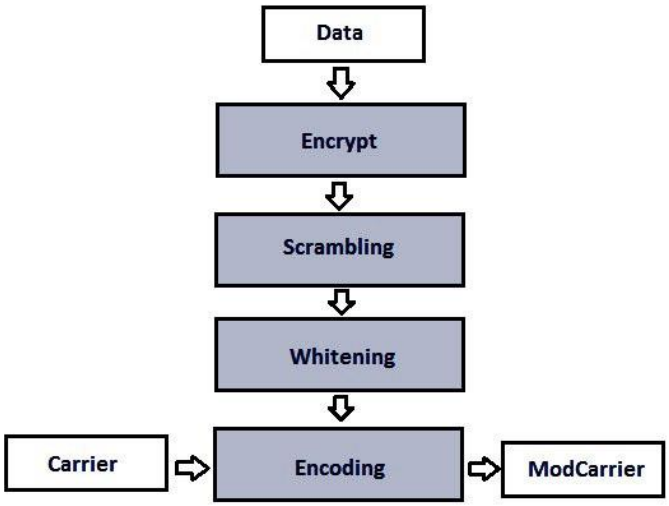
Các lớp cụ thể như sau:

- Layer 1 - Modern Multi-Cryptography: Một bộ 16 thuật toán mật mã hiện đại mã nguồn mở 256 bit đã được kết hợp thành thuật toán đa mật mã, sử dụng hai mật khẩu (256 bit + 256 bit).

- Layer 2 - CSPRNG Based Scrambling: Dữ liệu đã bị mã hóa sẽ bị xáo trộn để phá vỡ bất kỳ luồng bit nào còn tồn tại. Sử dụng một bộ Cryptographically Secure Pseudorandom Number Generator (CSPRNG), hay còn gọi là bộ tạo số giả ngẫu nhiên mật mã, được gieo với một mật khẩu thứ 3 (256 bit) và dữ liệu được xáo trộn hoàn toàn với những chỉ số ngẫu nhiên.

- Layer 3 - CSPRNG Based Whitening: Dữ liệu đã bị xáo trộn sẽ được làm trắng bằng cách trộn lẫn với một lượng lớn nhiễu, được lấy từ một bộ CSPRNG độc lập được gieo cùng với một Entropy phần cứng.

- Layer 4 - Adaptive Non-Linear Encoding: Dữ liệu đã làm trắng được mã hóa bằng cách sử dụng một hàm phi tuyến tính, sử dụng các bit của đối tượng vận chuyển ban đầu làm đầu vào. Những đối tượng vận chuyển đã được sửa đổi sẽ cần ít thay đổi hơn trong quá trình nhúng dữ liệu bí mật và đánh lừa được nhiều thử nghiệm Steganalysis.

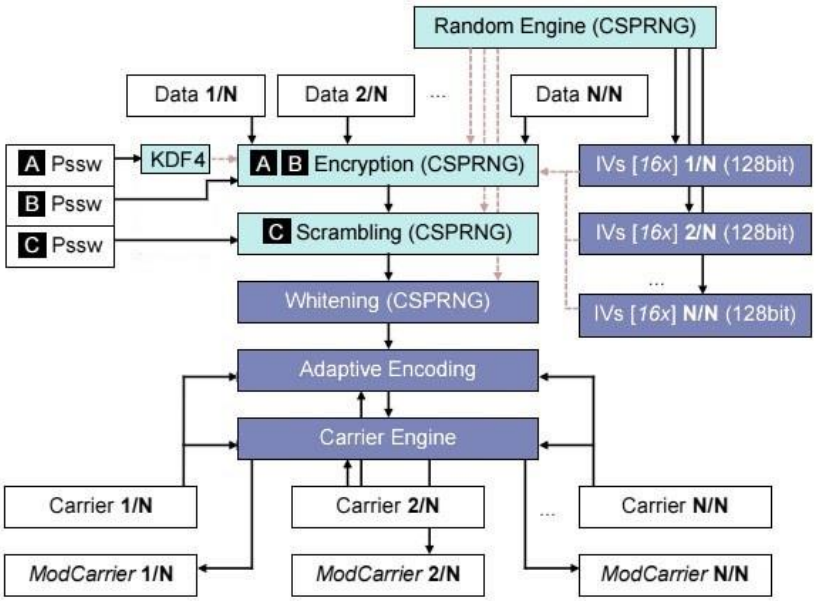


Hình 2.1. Các lớp trong mô hình bảo mật 4 lớp.

Ngoài ra EmbeddedSW Company còn phát triển thêm chức năng "bảo mật thêm" hay còn gọi là Deniable Steganography cho mô hình này. Dữ liệu bí mật hàng đầu có thể được bảo vệ bằng cách sử dụng dữ liệu ít bí mật hơn như Decoy (mồi nhử) nó được xem như là dữ liệu "bí mật thay thế" trong trường hợp người giấu tin buộc phải tiết lộ dữ liệu bí mật của mình [4].

2.1.2. Kiến trúc của mô hình bảo mật 4 lớp

Mô hình bảo mật 4 lớp được xây dựng với kiến trúc như hình vẽ sau:



Hình 2.2. Kiến trúc Steganography của mô hình bảo mật 4 lớp.

Với cơ chế hoạt động:

- Dữ liệu được phân chia thành nhiều khối với kích thước nhất định: Data 1/N, Data 2/N,…, Data N/N.

- Có thể sử dụng nhiều đối tượng vận chuyển, số lượng đối tượng vận chuyển phụ thuộc vào lượng dữ liệu cần giấu.

- Một mảng Vector khởi tạo ngẫu nhiên IV được kết hợp trong quá trình mã hóa dữ liệu.

- Mật khẩu văn bản (32 ký tự = 256 bit) được mở rộng thành 16 mật khẩu phục vụ cho quá trình mã hóa.

- Dữ liệu được mã hóa lần đầu với hai khóa 256 bit (A, B), một khóa sử dụng cho bộ tạo số giả ngẫu nhiên mật mã CSPRNG, một khóa sử dụng cho quá trình mã hóa Multi-Cryptography.

- Dữ liệu đã mã hóa sau đó được xáo trộn, cùng với khóa thứ 3 (C), để phá vỡ bất cứ luồng bit nào còn lại.

- Dữ liệu đã xáo trộn sau đó được làm trắng bằng việc trộn nhiễu ngẫu nhiên.

- Dữ liệu đã làm trắng lại tiếp tục được mã hóa sử dụng một phương thức mã hóa phi tuyến tính sử dụng các bit vận chuyển nguyên mẫu làm đầu vào. Nhằm mục đích chỉnh sửa các đối tượng vận chuyển trước khi nhúng dữ liệu vào chúng để có thể vượt qua các thử nghiệm Steganalysis.

- Cuối cùng các đối tượng vận chuyển đã được chỉnh sửa nhận các luồng xử lý. Từng khối dữ liệu đã qua các quá trình mã hóa trên được nhúng vào các đối tượng vận chuyển thông qua một bộ Carrier Engine [4].

Như vậy, nội dung trên đã trình bày tổng quan về mô hình 4 lớp và cơ chế hoạt động của nó. Còn về phần kiến trúc từng lớp sẽ được trình bày chi tiết trong các nội dung tiếp theo.

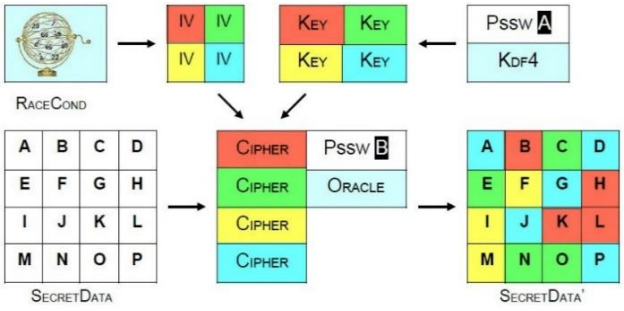
2.2. Chức năng các lớp trong mô hình bảo mật 4 lớp

2.2.1. Layer 1 - Modern Multi-Cryptography

Layer 1 - Modern Multi-Cryptography được xây dựng với việc sử dụng một bộ gồm 16 thuật toán mã hóa hiện đại mã nguồn mở. Các thuật toán mã hóa sử dụng trong lớp này được chọn từ các thuật toán: *AES Process* [1997-2000], *NESSIE Process* [2000-2003] và *CRYPTREC Process* [2000-2003].

Số thuật toán đó được kết hợp lại thành một thuật toán đa mật mã sử dụng 2 mật khẩu với kích thước 256 bit. Cụ thể Multi-Cryptography bao gồm các thuật toán mã hóa sau: *AES*, *Anubis*, *Camellia*, *Cast-256*, *Clefia*, *FROG*, *Hierocrypt3*, *Idea-NXT*, *MARS*, *RC6*, *Safer+*, *SC2000*, *Serpent*, *Speed*, *Twofish*, *Unicorn-A* [4].

Mô hình của Layer 1 được trình bày như hình vẽ sau:



Hình 2.3. Sơ đồ mô hình Layer 1 - Modern Multi-Cryptography.

Ở sơ đồ trên mô tả quá trình mã hóa dữ liệu bí mật bằng cách sử dụng một bộ đa mật mã được chọn lựa ngẫu nhiên, được kết hợp với bộ khóa Key được mở rộng nhờ hàm KDF4 và mảng Vector khởi tạo IV.

KDF4 viết tắt của Key Derivation Function 4 là một hàm có chức năng cung cấp thêm nhiều khóa bí mật từ một khóa chủ nhằm phục vụ cho quá trình tạo số giả ngẫu nhiên mật mã CSPRNG. KDF4 trong mô hình bảo mật 4 lớp sử dụng 4 thuật toán băm mật mã 512 bit là: *Grostl, Keccak, SHA2, Skein*.

Layer 1 chính là một phần của hệ thống mã nguồn mở LibObfuscate đã được phát triển trước đó. Ở đây, dữ liệu sẽ được qua một bộ mã hóa Multi-Cryptography 128 bit block - 256 bit Keys [16x] - CBC.

Multi-Cryptography sử dụng hai mật khẩu 256 bit (A và B). Mật khẩu thứ nhất (A) sẽ được mở rộng bằng hàm KDF4, và các mật khẩu đã được mở rộng đó sẽ được cung cấp cho 16 thuật toán mã hóa đã nêu để thực hiện quá trình mã hóa dữ liệu, nhờ sử dụng 1 bộ CSPRNG để thực hiện quá trình chọn ngẫu nhiên thuật toán mã hóa. Cụ thể thuật toán mã hóa Multi-Cryptography sẽ được trình bày chi tiết ở phần thuật toán liên quan.

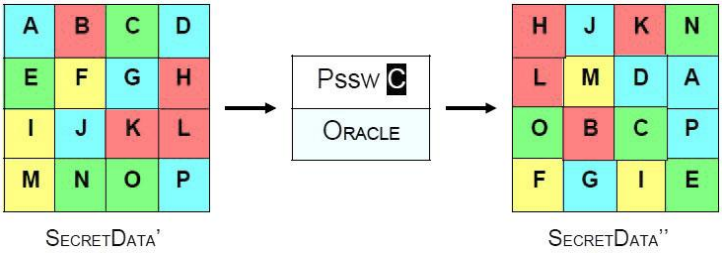
Như vậy, đây là lớp có vai trò rất quan trọng trong mô hình 4 lớp. Dữ liệu đi qua lớp này sẽ từ bản rõ được mã hóa hoàn toàn thành các dữ liệu không thể dễ dàng để có thể hiểu được. Các lớp còn lại của mô hình sẽ được trình bày ở các phần tiếp theo.

2.2.2. Layer 2 - CSPRNG Based Scrambling

Trong Layer 2 - CSPRNG Based Scrambling, các dữ liệu đã mã hóa luôn phải được xáo trộn để phá vỡ bất kỳ cấu trúc của các luồng bit còn lại. Layer 2 sử dụng một bộ CSPRNG với một mật khẩu thứ ba 256 bit và các dữ liệu được trộn lẫn hoàn toàn với các chỉ số ngẫu nhiên.

Ở lớp này việc xáo trộn được thực hiện bằng thuật toán Scrambling (xáo trộn) biến đổi thứ tự các bit của dữ liệu đầu vào thành các bít đầu ra có thứ tự ngẫu nhiên nhờ sử dụng bộ CSPRNG được gieo cùng với một mật khẩu thứ 3 (C). Mật khẩu (C) đóng vai trò khởi tạo Seed cho quá trình tạo số ngẫu nhiên sử dụng CSPRNG. Dữ liệu được xáo trộn hoàn toàn ngẫu nhiên nên việc có thể sắp xếp lại nó mà không cần mật khẩu là điều không thể.

Scrambling cũng là một phần trong hệ thống mã nguồn mở LibObfuscate với mô hình tổng quan được thể hiện như hình sau [4].

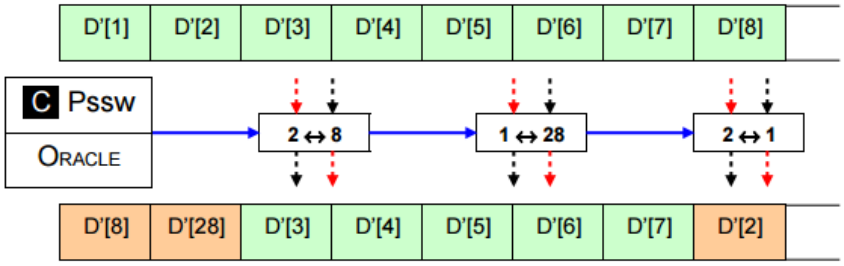


Hình 2.4. Sơ đồ mô hình của Layer 2 - CSPRNG Based Scrambling.

Với hoạt động như sau:

- CSPRNG được thiết lập với một mật khẩu độc lập (C).

- Cho một luồng đầu vào n byte, thực hiện n/2 lần thuật toán xáo trộn ngẫu nhiên In-place, là một thuật toán biến đổi đầu vào không sử dụng cấu trúc dữ liệu phụ trợ, không hạn chế về việc lặp lại chỉ số. Với mô tả như hình dưới [6]:



Hình 2.5. Mô tả thuật toán Scrambling trong Layer 2 - CSPRNG Based Scrambling.

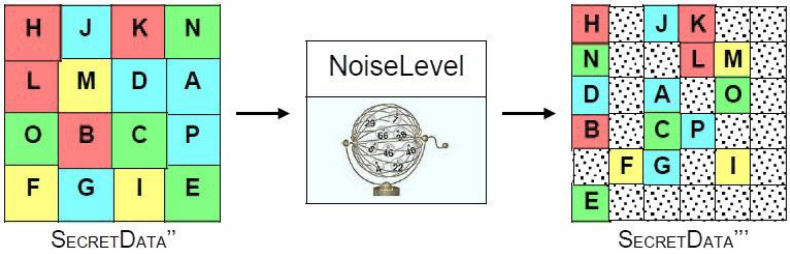
Thuật toán Scrambling sẽ được kết hợp với một bộ CSPRNG được gieo cùng với một mật khẩu (C) để tạo một mảng các cặp chỉ số ngẫu nhiên n/2 phần tử, với n là độ dài khối của dữ liệu vào. Sau đó thực hiện đổi chỗ các khối theo mảng các cặp chỉ số đã được tạo để tiến hành xáo trộn dữ liệu ban đầu.

2.2.3. Layer 3 - CSPRNG Based Whitening

Ở Layer 3 - CSPRNG Based Whitening, các dữ liệu đã bị xáo trộn sẽ được trộn với một lượng lớn nhiễu, tạo ra bởi một CSPRNG độc lập được khởi tạo với một Entropy phần cứng.

Trong mô hình bảo mật 4 lớp thì Layer 3 là một công nghệ độc quền. Nên trong nội dung nghiên cứu của đề tài chỉ đề cập ở mức độ giới thiệu tổng quan về lớp này chứ không thể nghiên cứu chuyên sâu.

Layer 3 có mô hình như sau [4]:



Hình 2.6. Sơ đồ mô hình của Layer 3 - CSPRNG Based Whitening.

Trong đó:

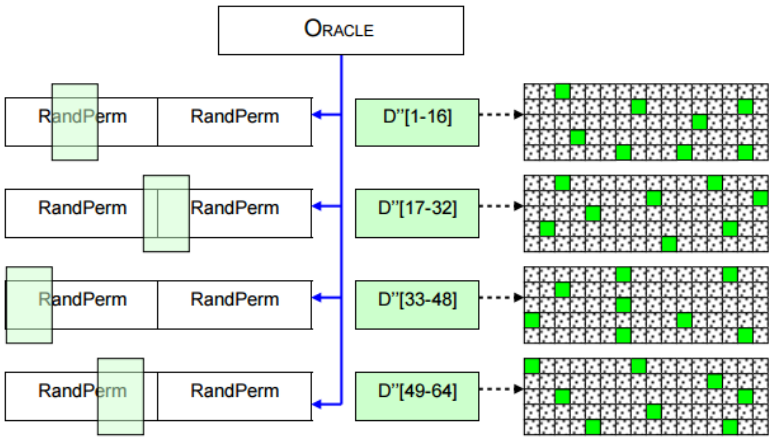
- Dữ liệu và nhiễu (phụ thuộc chính vào mức độ nhiễu) được trộn vào các khối có kích thước cố định. Giả sử các khối dữ liệu có kích thước là 960 byte, ta có:

+ Minimum: 300% nhiễu (720 byte) / dữ liệu (240 byte).

+ Maximum: 5900% nhiễu (944 byte) / dữ liệu (16 byte).

- Kích thước khối 960 byte buộc những kẻ tấn công sẽ phải kiểm tra tất cả những khối nhiễu khả dụng.

- Sử dụng CSPRNG kết hợp với thuật toán hoán vị Bit-level Durstenfeld’s-Shuffled P-Box, cung cấp một công cụ làm trắng với một tập hợp các chỉ số ngẫu nhiên không lặp.



Hình 2.7. Mô tả thuật toán Whitening của Layer 3 - CSPRNG Based Whitening.

Ở thuật toán này, CSPRNG được sử dụng để tạo các chỉ số ngẫu nhiên không lặp, và được trộn với các khối dữ liệu nhiễu một cách ngẫu nhiên.

Whitening cũng là thành phần chính của Deniable Encryption, hoặc Deniable Steganography, chức năng "bảo mật thêm" của mô hình:

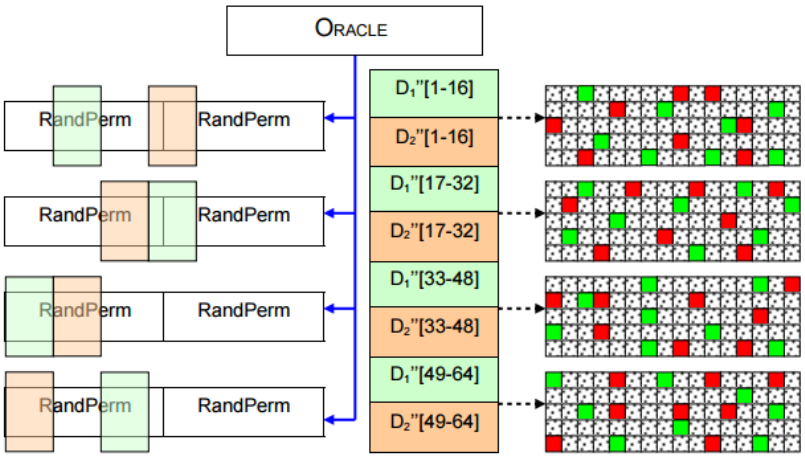
- Mô hình bảo mật 4 lớp hỗ trợ ẩn dữ liệu bí mật và nhiều Decoy, n mức Deniable Steganography, được gọi là các Aspect.

- Số Aspect tối đa phụ thuộc ở mức lõi LibObfuscate, về mức độ nhiễu với kích thước khối là 960 byte:

+ Minimum: 4 Aspect ứng với 300% nhiễu/dữ liệu.

+ Maximum: 60 Aspect ứng với 5900% nhiễu/dữ liệu.

Việc chèn Decoy cũng tượng tự việc chèn dữ liệu bí mật, nhưng chúng chỉ được chèn lên những vị trí trống chưa được sử dụng, không thể chèn đè lên các dữ liệu bí mật [6].



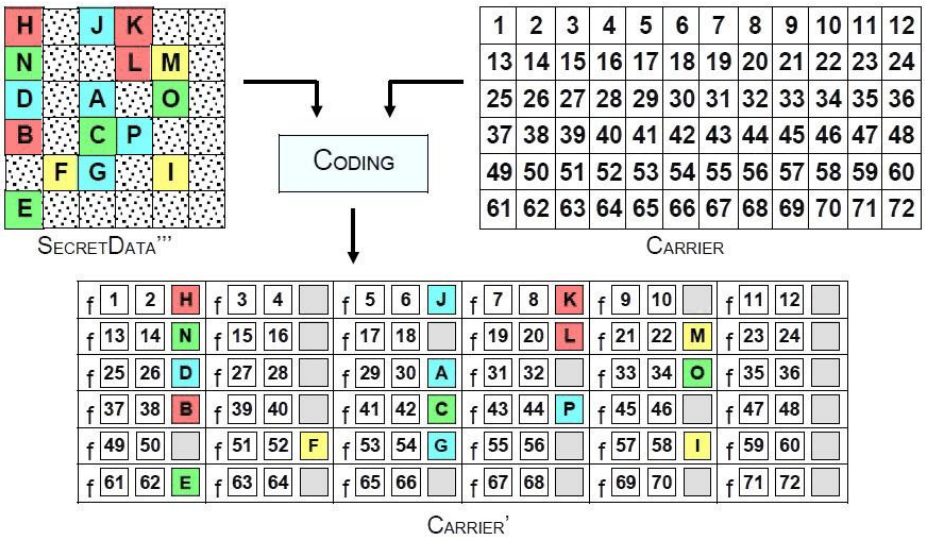
Hình 2.8. Quá trình chèn Decoy ngẫu nhiên nhằm mục đích Deniable Steganography.

2.2.4. Layer 4 - Adaptive Non-Linear Encoding

Ở Layer 4 - Adaptive Non-Linear Encoding, các dữ liệu bị làm trắng luôn được mã hóa tiếp bằng cách sử dụng chức năng phi tuyến tính, sử dụng các bit vận chuyển nguyên mẫu làm đầu vào, nhằm mục đích vượt qua nhiều thử nghiệm Steganalysis.

Đây cũng là một lớp độc quyền của mô hình bảo mật 4 lớp. Vì vậy các tài liệu chuyên sâu về lớp này không được chia sẻ. Do đó đề tài cũng chỉ dừng lại ở mức độ giới thiệu tổng quan như ở Layer 3.

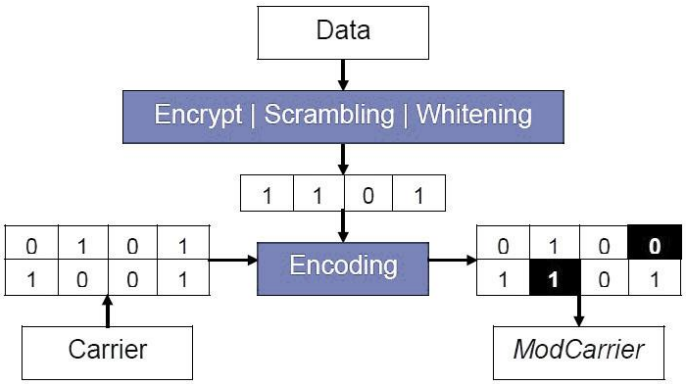
Layer 4 có sơ đồ như sau:



Hình 2.9. Sơ đồ mô hình của Layer 4 - Adaptive Non-Linear Encoding.

Dữ liệu trước khi tiêm vào các tập tin vận chuyển, đã được mã hóa và làm trắng nên một lượng nhỏ dữ liệu bí mật cần ẩn sẽ trở thành một lượng dữ liệu ngẫu nhiên lớn hơn, sẽ gây sự chú ý và nghi ngờ. Vì vật các tập tin vận chuyển trước khi được tiêm cần mã hóa nó bằng cách sử dụng một hàm phi tuyến tính lấy các bit của các tập tin vận chuyển nguyên mẫu như đầu vào. Các tập tin vận chuyển đã được sửa đổi sẽ chỉ cần thay đổi ít hơn và làm giảm lượng phản hồi thống kê của chúng, đánh lừa nhiều bài kiểm tra Steganalysis [4].

Ta có ví dụ mã hóa ở Layer 4 như hình vẽ sau:



Hình 2.10. Ví dụ về quá trình mã hóa của Layer 4 - Adaptive Non-Linear Encoding.

Ở ví dụ trên, để tiến hành tiêm các bít Data 1101 vào Carrier có dãy bit 0101 1001. Ta sẽ thực hiện quá trình mã hóa Carrier bằng hàm phi tuyến tính thành 0100 1101 để chứa dữ liệu bí mật là 1101, đồng thời số lượng bit 0 và bit

1 của ModCarrier vẫn không thay đổi nên có thể vượt qua các bài kiểm tra thống kê.

Nội dung trên đã trình bày về các lớp trong mô hình bảo mật 4 lớp. Tiếp theo, các phần ngay sau đây sẽ nghiên cứu, tìm hiểu về các thuật toán được sử dụng trong mô hình này.

2.3. Các thuật toán được sử dụng trong mô hình bảo mật 4 lớp

Trong mô hình bảo mật 4 lớp được sử dụng rất nhiều thuật toán như:

- CSPRNG - Based on AES.

- Hashing - 512 bit: Grostl, Keccak, SHA2, Skein.

- Cryptography - 256 bit - ECB - 128 bit block: AES, Anubis, Camellia, Cast-256, Clefia, FROG, Hierocrypt3, Idea-NXT, MARS, RC6, Safer+, SC2000, Serpent, Speed, Twofish, Unicorn-A, Wrapper.

- Multi-Cryptography: 256 + 256 bit - CBC - Segment.

Nhưng ở phạm vi đồ án này chỉ tập trung nghiên cứu một số thuật toán nổi bật trong mô hình bảo mật 4 lớp.

2.3.1. Thuật toán Cryptographically Secure Pseudo-Random Number Generator dựa trên AES - CTR

2.3.1.1. Thuật toán Cryptographically Secure Pseudo-Random Number Generator

Cryptographically Secure Pseudo-Random Number Generator (CSPRNG) hoặc Cryptographic Pseudo-Random Number Generator (CPRNG) là một Pseudo-Random Number Generator (PRNG) được kết hợp với các thuộc tính riêng biệt làm cho nó phù hợp hơn để sử dụng trong mật mã học.

Trong đó, PRNG là một thuật toán nhằm mục đích tạo ra một dãy số có các thuộc tính của chuỗi các số ngẫu nhiên.

Mặc dù chuỗi số do PRNG tạo ra không thực sự ngẫu nhiên, vì nó được xác định bởi một tập các giá trị ban đầu tương đối nhỏ, được gọi là hạt giống PRNG. Nhưng bộ tạo số ngẫu nhiên rất quan trọng trong thực tế bởi vì tốc độ của chúng trong quá trình tạo số và khả năng tái sản xuất.

Nhiều khía cạnh của mật mã học đòi hỏi số ngẫu nhiên như:

- Key Generation: Là quá trình tạo ra khóa trong Cryptography. Một khóa được dùng để mã hóa và giải mã bất cứ dữ liệu nào được mã hóa hoặc

giải mã. Một thiết bị hoặc phần mềm sử dụng những Generation Key được gọi là Key Generator hoặc Keygen.

- Nonce: Là một số tùy ý mà chỉ được sử dụng một lần. Nó thường là một số ngẫu nhiên hoặc giả ngẫu nhiên được phát hành trong một giao thức xác thực để đảm bảo rằng truyền thông cũ không thể được tái sử dụng trong các cuộc tấn công Replay Attack. Chúng cũng có thể hữu ích như các Vector khởi tạo và hàm băm mật mã.

- One-Time Pad: Là một kỹ thuật mã hóa mà không thể bị phá. Trong kỹ thuật này, một bản rõ được ghép nối với một khóa bí mật ngẫu nhiên. Khi đó, mỗi bit hoặc ký tự của bản rõ được mã hóa bằng cách kết hợp nó với các bit hoặc ký tự tương ứng của Pad sử dụng mô-đun số học.

- Salt trong lược đồ chữ ký số: Là dữ liệu ngẫu nhiên được sử dụng như một đầu vào bổ sung cho một chức năng một chiều để băm một mật khẩu hoặc một cụm mật khẩu.

Một CSPRNG phải đáp ứng được cả hai yêu cầu sau:

- Đầu tiên, chúng phải vượt qua các bài kiểm tra thống kê ngẫu nhiên.

- Thứ 2, chúng chống đỡ tốt dưới các cuộc tấn công nguy hiểm, ngay cả khi một phần của trạng thái ban đầu hoặc hoạt động của chúng trở nên khả dụng cho kẻ tấn công.

Cụ thể:

- Mỗi CSPRNG phải đáp ứng các bài kiểm tra Next-bit. Đó là, khi cho k bit đầu tiên của một chuỗi ngẫu nhiên, không có thuật toán thời gian đa thức nào mà có thể dự đoán bit thứ k+1 với xác suất thành công lớn hơn 50%. Andrew Yao, một nhà khoa học máy tính Trung Quốc, đã chứng minh vào năm 1982 rằng một hệ thống sinh ngẫu nhiên vượt qua bài kiểm tra Next-bit sẽ vượt qua tất cả các bài kểm tra thống kê thời gian đa thức khác cho ngẫu nhiên.

- Mỗi CSPRNG phải chống được các State Compromise Extension, những phần mở rộng trạng thái gây hại. Trong trường hợp một phần hoặc toàn bộ trạng thái của nó đã được tiết lộ hoặc đoán chính xác. Nó không nên tái tạo lại chuỗi số ngẫu nhiên trước khi để tiết lộ.

Ngoài ra, nếu có một đầu vào Entropy trong khi hoạt động, nó sẽ không khả thi khi sử dụng hiểu biết về trạng thái đầu vào để có thể đoán được các điều kiện tương lai của trạng thái CSPRNG.

Phần này chỉ giới thiệu về thuật toán CSPRNG để tạo cơ sở cho các phần được trình bày sau đó.

2.3.1.2. Thuật toán AES

AES viết tắt của từ Advanced Encryption Standard, hay tiêu chuẩn mã hóa tiên tiến, là một thuật toán mã hóa khối được chính phủ Hoa kỳ áp dụng làm tiêu chuẩn mã hóa.

AES giống như tiêu chuẩn tiền nhiệm DES, nó cũng được kỳ vọng sẽ áp dụng trên phạm vi thế giới và đã được nghiên cứu rất kỹ lưỡng. AES được chấp thuận làm tiêu chuẩn liên bang bởi Viện Tiêu chuẩn và Công nghệ quốc gia Hoa Kỳ (NIST) sau một quá trình tiêu chuẩn hóa kéo dài 5 năm.

Thuật toán AES được thiết kế bởi hai nhà mật mã học người Bỉ là: Joan Daemen và Vincent Rijmen. Thuật toán này được đặt tên là Rijndael khi tham gia cuộc thi thiết kế AES. Mặc dù 2 tên AES và Rijndael vẫn thường được gọi thay thế cho nhau nhưng trên thực tế thì 2 thuật toán không hoàn toàn giống nhau.

AES chỉ làm việc với các khối dữ liệu đầu vào và đầu ra có kích thước 128 bit và khóa có độ dài 128, 192 hoặc 256 bit. Trong khi Rijndael có thể làm việc với dữ liệu và khóa có độ dài bất kỳ là bội số của 32 bit, nằm trong khoảng từ 128 đến 256 bit. Các khóa con sử dụng trong các chu trình được tạo ra bởi quá trình tạo khóa con Rijndael. Mỗi khóa con cũng là một cột gồm 4 byte.

Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu 128 bit đầu vào được chia thành 16 byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay là một ma trận 4x4 của các byte, nó được gọi là ma trận trạng thái. Trong quá trình thực hiện thuật toán các toán tử tác động để biến đổi ma trận trạng thái này.

Quá trình thực hiện thuật toán AES có 2 công việc:

- Thứ nhất là mở rộng khóa (Key Expansion): Là quá trình tạo các vòng khóa từ khóa chính, mỗi khóa con chứa 4 byte.

- Thứ hai là quá trình mã hóa. Bao gồm các bước:

1. Khởi động vòng lặp:

1. AddRoundKey - Mỗi cột của trạng thái đầu tiên lần lượt được kết hợp với một khóa con theo thứ tự từ đầu dãy khóa.

2. Vòng lặp:

1. SubBytes - Đây là phép thế (phi tuyến) trong đó mỗi byte trong trạng thái sẽ được thế bằng một byte khác theo bảng tra (Rijndael S-box).

2. ShiftRows - Dịch chuyển, các hàng trong trạng thái được dịch vòng theo số bước khác nhau.

3. MixColumns - Quá trình trộn làm việc theo các cột trong khối theo một phép biến đổi tuyến tính.

4. AddRoundKey

3. Vòng lặp cuối:

1. SubBytes

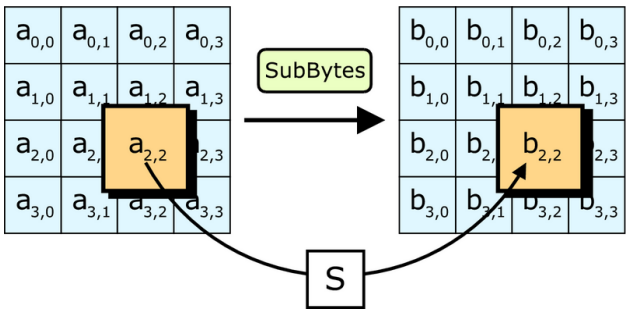
2. ShiftRows

3. AddRoundKey

Tại chu trình cuối thì bước MixColumns không thực hiện.

Cụ thể thì các bước được thực hiện như sau:

- *Bước SubBytes:* Các byte được thế thông qua bảng tra S-box. Đây chính là quá trình phi tuyến của thuật toán. Hộp S-box này được tạo ra từ một phép biến đổi khả nghịch trong trường hữu hạn GF (28 bit) có tính chất phi tuyến. Để chống lại các tấn công dựa trên các đặc tính đại số, hộp S-box này được tạo nên bằng cách kết hợp phép nghịch đảo với một phép biến đổi Affine khả nghịch. Hộp S-box này cũng được chọn để tránh các điểm bất động. Trong bước SubBytes, mỗi byte được thay thế bằng một byte theo bảng tra S: $b_{ij} = S(a_{ij})$.

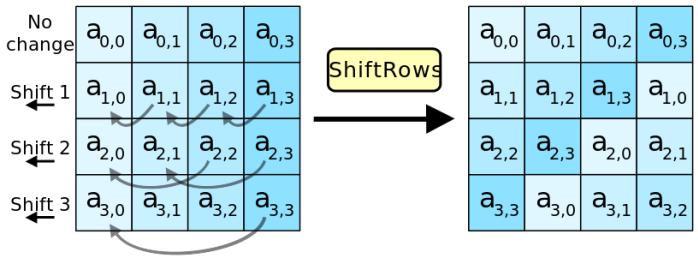


Hình 2.11. Mô hình của bước SubBytes.

- *Bước ShiftRows:* Các hàng được dịch vòng một số bước nhất định. Đối với AES, hàng đầu được giữ nguyên. Mỗi byte của hàng thứ 2 được dịch vòng trái một vị trí. Tương tự, các hàng thứ 3 và 4 được dịch vòng 2 và 3 vị trí. Do

vậy, mỗi cột khối đầu ra của bước này sẽ bao gồm các byte ở đủ 4 cột khối đầu vào.

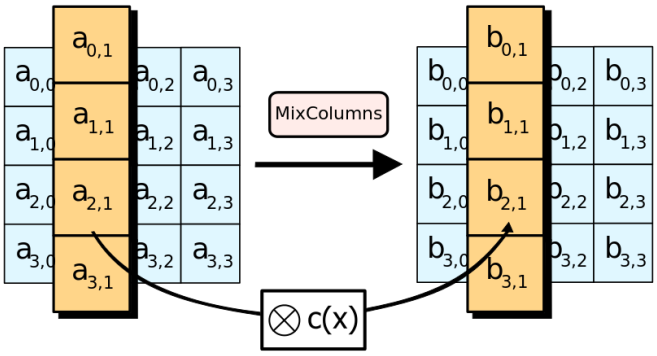
Đối với Rijndael với độ dài khối khác nhau thì số vị trí dịch chuyển cũng khác nhau. Trong bước ShiftRows, các byte trong mỗi hàng được dịch vòng trái. Số vị trí dịch chuyển tùy thuộc từng hàng.



Hình 2.12. Mô hình của bước ShiftRows.

- *Bước MixColumns:* Bốn byte trong từng cột được kết hợp lại theo một phép biến đổi tuyến tính khả nghịch. Mỗi khối 4 byte đầu vào sẽ cho một khối 4 byte ở đầu ra với tính chất là mỗi byte ở đầu vào đều ảnh hưởng tới cả 4 byte đầu ra.

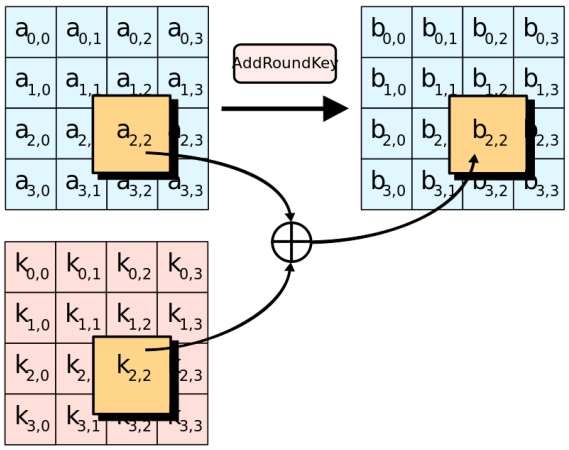
Cùng với bước ShiftRows, MixColumns đã tạo ra tính chất khuếch tán cho thuật toán. Mỗi cột được xem như một đa thức trong trường hữu hạn và được nhân với đa thức $c(x) = 3x^3 + x^2 + x + 2$ (modulo $x^4 + 1$). Vì thế, bước này có thể được xem là phép nhân ma trận trong trường hữu hạn. Trong bước MixColumns, mỗi cột được nhân với một hệ số cố định c(x).



Hình 2.13. Mô hình của bước MixColumns.

- *Bước AddRoundKey:* Tại bước này, khóa con được kết hợp với các khối. Khóa con trong mỗi chu trình được tạo ra từ khóa chính với quá trình tạo khóa con Rijndael; mỗi khóa con có độ dài giống như các khối.

Quá trình kết hợp được thực hiện bằng cách XOR từng bít của khóa con với khối dữ liệu. Trong bước AddRoundKey, mỗi byte được kết hợp với một byte trong khóa con của chu trình sử dụng phép toán XOR.



Hình 2.14. Mô hình của bước AddRoundKey.

- *Tối ưu hóa:* Đối với các hệ thống 32 bít hoặc lớn hơn, ta có thể tăng tốc độ thực hiện thuật toán bằng cách sát nhập các bước SubBytes, ShiftRows, MixColumns và chuyển chúng thành dạng bảng. Có cả thảy 4 bảng với 256 mục, mỗi mục là 1 từ 32 bít, 4 bảng này chiếm 4096 byte trong bộ nhớ. Khi đó, mỗi chu trình sẽ được bao gồm 16 lần tra bảng và 12 lần thực hiện phép XOR 32 bít cùng với 4 phép XOR trong bước AddRoundKey.

Trong trường hợp kích thước các bảng vẫn lớn so với thiết bị thực hiện thì chỉ dùng một bảng và tra bảng kết hợp với hoán vị vòng quanh [11].

2.3.1.3. Thuật toán Counter Mode

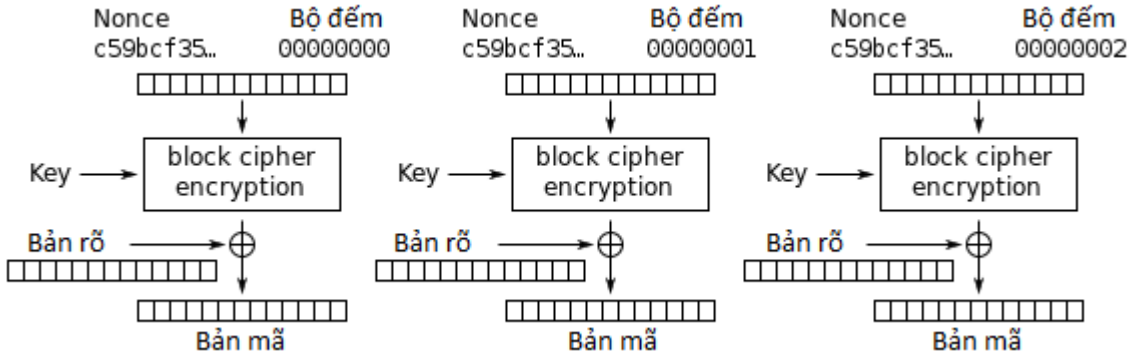
Counter Mode (CTR Mode) còn được gọi là chế độ truy cập số nguyên hoặc chế độ đếm số nguyên phân đoạn, được chuẩn hóa bởi NIST trong SP 800-38A vào năm 2001.

CTR Mode sử dụng một bộ đếm thay vì một bộ IV truyền thống. Bộ đếm có các thuộc tính bổ sung bao gồm: Một Nonce và một bộ đếm ban đầu. Ở chế độ này không yêu cầu phải đệm thêm vào các bản rõ để đạt kích thước khối của bản mã.

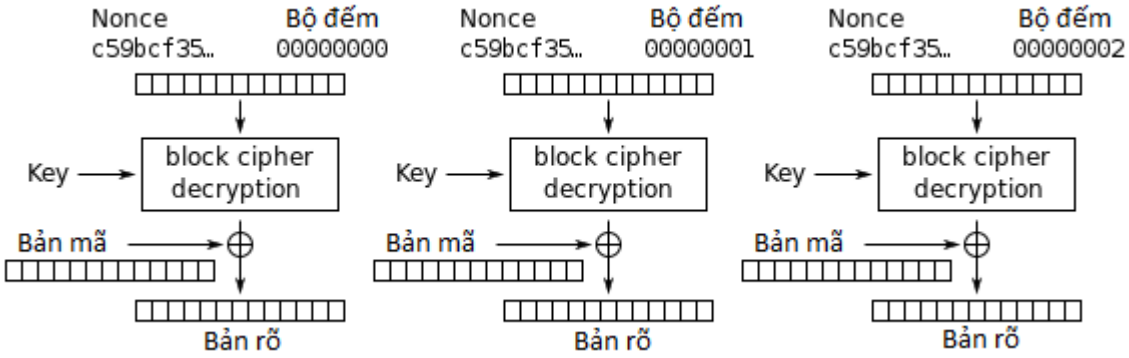
CTR Mode có chức năng biến một mật mã khối thành một mật mã dòng. Nó tạo ra khối Keystream tiếp theo bằng cách mã hóa các giá trị kế tiếp của một bộ đếm. Bộ đếm phải tạo ra một chuỗi trình tự đảm bảo không lặp lại trong một thời gian dài. CTR Mode thích hợp để vận hành trên một máy đa xử lý, nơi các khối có thể được mã hóa song song.

Trong trường hợp Nonce được chọn ngẫu nhiên, chúng có thể được kết hợp với bộ đếm bằng cách sử dụng bất kỳ phép toán không làm mất dữ liệu nào như: Phép nối, thêm hoặc XOR để tạo ra khối bộ đếm duy nhất cho mã hóa. Trong trường hợp None không ngẫu nhiên, chẳng hạn như một bộ đếm gói tin, các None và bộ đếm được nối lại. Ví dụ, lưu trữ Nonce trong 64 bit cao và bộ đếm trong 64 bit thấp của khối bộ đếm 128 bit [12].

Sơ đồ dưới đây mô tả quá trình mã hóa và giải mã sử dụng CTR Mode:



Hình 2.15. Sơ đồ mã hóa của Counter Mode.



Hình 2.16. Sơ đồ giải mã của Counter Mode.

Do tính đối xứng của phép toán XOR, mã hóa và giải mã thực chất được thực hiện các bước giống nhau.

Ta quy ước:

- C_i là khối bản mã thứ i.

- P_i là khối bản rõ thứ i.

- O_i là khối mã hóa thứ i.

- I_i là bộ đếm thứ i, bao gồm phần None và phần Counter.

- K là khóa.

- E là thuật toán mã hóa.

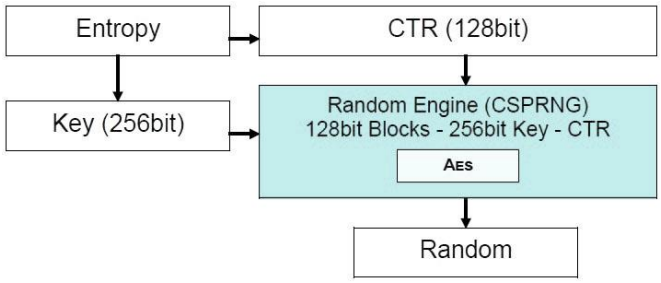
Ta có công thức mã hóa: $C_i = P_i$ XOR O_i.

Ta có công thức giải mã: $P_i = C_i$ XOR O_i.

Trong đó: $O_i = E_k (I_i)$; $I_i = I_{i-1} + 1$.

2.3.1.4. Sử dụng thuật toán Cryptographically Secure Pseudo-Random Number Generator dựa trên AES - CTR trong mô hình bảo mật 4 lớp

Trong mô hình bảo mật 4, một bộ CSPRNG sử dụng mã hóa AES-256 và các mật mã khối an toàn được thực thi trong CTR Mode có cấu trúc như sơ đồ sau:



Hình 2.17. Sơ đồ cấu trúc thuật toán Cryptographically Secure Pseudo-Random Number Generator.

Sơ đồ trên là một bộ CSPRNG tạo số ngẫu nhiên nhờ xử lý các khối dữ liệu đầu vào 128 bit với khóa có kích thước 256 bit dựa trên mã hóa AES được thực hiện trong CTR Mode. Nói một cách dễ hiểu thì nó chính là CTR Mode được giải thích ở phần trước với thuật toán mã hóa E là AES-256 bit, có kích thước bộ đếm là 128 bit.

Thuật toán CSPRNG dựa trên AES - CTR có 2 phần chính [13]:

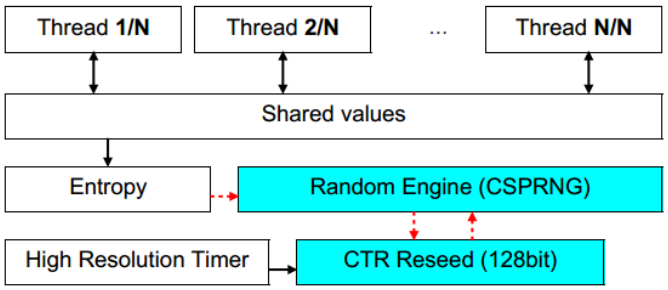
- Thuật toán tạo Seed: Seed là yếu tố quan trọng không thể thiếu trong bất kỳ bộ tạo số ngẫu nhiên nào. Trong thuật toán CSPRNG thì Seed được tạo bằng 2 cách:

+ Tạo Seed thủ công: Seed được tạo thủ công bằng hàm băm mật mã:

Seed = Hash (Passw, Nonce).

Trong đó: Hash là một trong số 4 hàm băm mật mã: *Grostl*, *Keccak*, *SHA2*, *Skein*; Passw là mật khẩu; Nonce là một số bất kỳ.

+ Tạo Seed tự động: Seed được tạo tự động bằng cách sử dụng các luồng xử lý của máy tính. Các luồng xử lý luôn được sắp xếp bởi hệ điều hành theo một thứ tự không thể đoán trước được. Giả sử có N luồng hoạt động song song, nó làm tăng hoặc giảm giá trị đã được chia sẻ trước đó, sau quá trình đó sẽ cho kết quả là các giá trị ngẫu nhiên.



Hình 2.18. Tạo Seed tự động bằng các luồng xử lý.

- Thuật toán Random: Thuật toán Random được sử dụng ở trong mô hình bảo mật 4 lớp thực chất là một thuật toán tạo số ngẫu nhiên bình thường được cải tiến để tăng tính bảo mật cho quá trình mã hóa. Cụ thể được mô tả như sau:

+ Tính ngẫu nhiên của hàm Random được khởi tạo nhờ các Seed ở trên.

+ Sử dụng thuật toán mã hóa AES với khối mã hóa 128 bit và mật khẩu 256 bit để mã hóa dựa trên CTR Mode.

+ Sau đó sẽ được lấy từng byte của chuỗi đầu ra để sử dụng cho các mục đích khác nhau trong thuật toán cần dùng đến số ngẫu nhiên.

Một số đặc điểm của thuật toán CSPRNG dựa trên AES-CTR trong mô hình 4 lớp là:

- Đầu vào của thuật toán là: Chuỗi Seed được tạo nhờ một mật khẩu 256bit và 1 số Nonce.

- Đầu ra của thuật toán là: Chuỗi số ngẫu nhiên được tạo nhờ mã hóa chuỗi Seed ban đầu nhờ sử dụng thuật toán AES-CTR.

- CSPRNG được sử dụng ở Layer 1 và Layer 2 có Seed được tạo nhờ sử dụng các khóa độc lập (B và C).

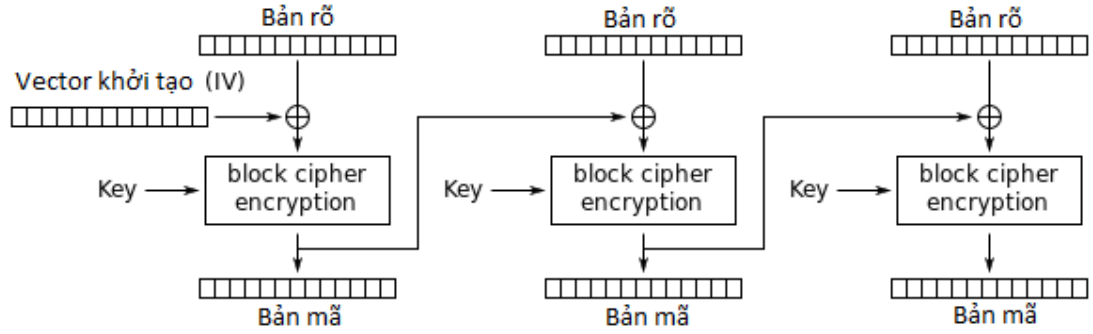
- CSPRNG được sử dụng ở Layer 3 có Seed được tạo tự động nhờ các luồng xử lý của hệ thống máy tính.

2.3.2. Thuật toán Multi-Cryptography dựa trên CBC Mode

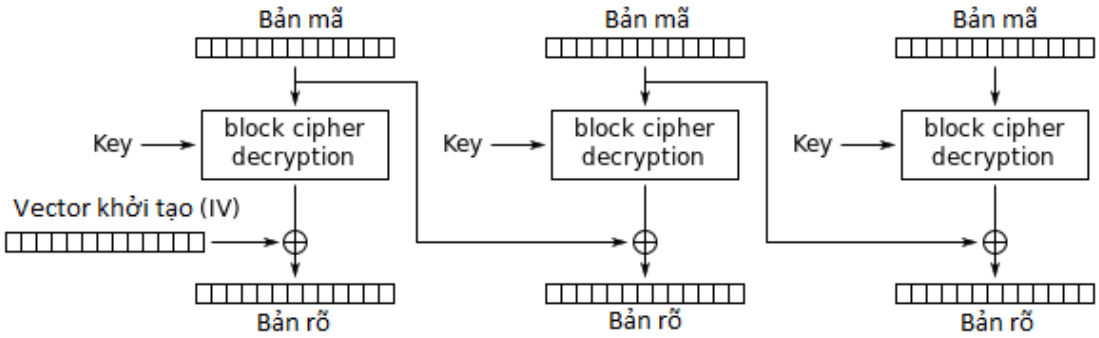
2.3.2.1. Thuật toán Cipher Block Chaining Mode

Cipher Block Chaining Mode (CBC Mode) là một chế độ hoạt động khác của mật mã khối, như CTR Mode. Nó được Ehrsam, Meyer, Smith và Tuchman phát minh vào năm 1976.

Trong CBC Mode, mỗi khối bản rõ được XOR với khối bản mã trước đó trước khi được mã hóa. Bằng cách này, mỗi khối bản mã phụ thuộc vào tất cả các khối bản rõ được xử lý đến điểm đó. Để làm cho mỗi một thông điệp là duy nhất, một IV phải được sử dụng trong khối đầu tiên.



Hình 2.19. Sơ đồ mã hóa của Cipher Block Chaining Mode.



Hình 2.20. Sơ đồ giải mã của Cipher Block Chaining Mode.

Nếu khối đầu tiên có chỉ số 1, công thức toán học cho mã hóa CBC là:

$C_i = E_k (P_i \text{ XOR } C_{i-1})$.

$C_0 = IV$.

Trong khi công thức toán học cho giải mã CBC là:

$P_i = D_k (C_i) \text{ XOR } C_{i-1.}$

$C_0 = IV$.

CBC là phương thức hoạt động phổ biến nhất. Những hạn chế chính của nó là mã hóa phải diễn ra tuần tự, và thông điệp phải được đệm đạt đến bội của

kích cỡ khối mã hóa. Lưu ý rằng một bit thay đổi trong một bãn rõ hoặc IV ảnh hưởng đến tất cả các khối bản mã phía sau.

Giải mã với IV không chính xác dẫn đến các khối đầu tiên của bản rõ sẽ bị hỏng nhưng các bản rõ sau đó vẫn sẽ đúng. Điều này là do mỗi khối được XOR với khối bản mã trước đó, không phải bản rõ. Do đó một lần giải mã không cần phải giải mã khối trước đó trước khi sử dụng nó như là IV cho lần giải mã hiện tại. Điều này có nghĩa là một khối bản rõ có thể được khôi phục từ hai khối bản mã liền kề. Do đó, việc giải mã có thể xảy ra song song.

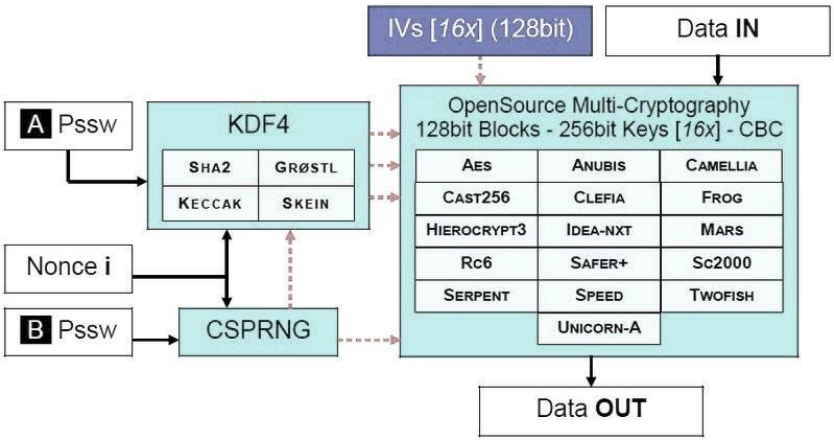
Lưu ý, một bit bị thay đổi cho bản mã có thể gây ra hư hỏng hoàn toàn các khối tương ứng của bản rõ, và đảo ngược các bit tương ứng trong khối theo sau của bản rõ, nhưng phần còn lại của các khối vẫn còn nguyên vẹn.

Các IV lợi dụng thuộc tính này bằng cách gộp trước một khối bản rõ ngẫu nhiên duy nhất. Mã hóa được thực hiện như bình thường, ngoại trừ các IV không cần phải được truyền đạt đến các giải mã thường lệ. Bất kể giải mã IV nào được sử dụng, chỉ có khối ngẫu nhiên là bị hỏng. Nó có thể được loại bỏ một cách an toàn và phần giải mã còn lại là bản rõ ban đầu [12].

2.3.2.2. Sử dụng Thuật toán Multi-Cryptography dựa trên CBC Mode trong mô hình bảo mật 4 lớp

Trong mô hình bảo mật 4 lớp, CBC Mode được lựa chọn trong Layer 1 với thuật toán mã hóa được sử dụng trong CBC Mode sẽ được chọn ngẫu nhiên nhờ CSPRNG trong số 16 thuật toán: AES, Anubis, Camellia, Cast-256, Clefia, FROG, Hierocrypt3, Idea-NXT, MARS, RC6, Safer+, SC2000, Serpent, Speed, Twofish, Unicorn-A, Wrapper.

Thuật toán này được gọi là Multi-Cryptography - 256 + 256 bit - CBC - Segment nhờ việc xử lý các khối mã hóa trong CBC Mode với nhiều thuật toán mã hóa cùng với hai khóa đầu vào có độ lớn 256 bit. Cụ thể thuật toán được thiết kế như sau:



Hình 2.21. Kiến trúc chi tiết của Multi-Cryptography.

Quá trình thiết lập Multi-Cryptography gồm 4 bước sau:

- Một mảng Vector khởi tạo ngẫu nhiên (16 x 128bit) được kết hợp với mỗi khối dữ liệu đầu vào.

- Một bộ Pseudo Random Engine (CSPRNG) được gieo sử dụng mật khẩu B.

- Mật khẩu A được mở rộng bằng KDF4 sử dụng 4 thuật toán băm hiện đại 512 bit mã nguồn mở, được chọn từ SHA2 và SHA3.

- Mỗi thuật toán băm tạo ra 4 khóa 256 bit:

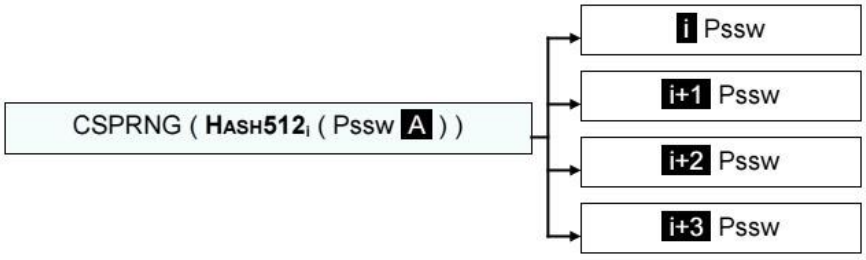
Pssw (1) | (2) | (3) | (4) = Rand (Sha2 (Pssw (A)))

Pssw (5) | (6) | (7) | (8) = Rand (Grøstl (Pssw (A)))

Pssw (9) | (10) | (11) | (12) = Rand (Keccak (Pssw (A)))

Pssw (13) | (14) | (15) | (16) = Rand (Skein (Pssw (A)))

Mảng khóa thu được kết hợp với mỗi mật mã sử dụng CSPRNG như hình sau:



Hình 2.22. Sử dụng CSPRNG để chọn khóa ngẫu nhiên.

Trong Multi-Cryptography, các mật mã mã hóa cũng là một quá trình đa bước:

- Mỗi dữ liệu có một thiết lập chung:

Setup = { { IV }, CSPRNG, { Key } }

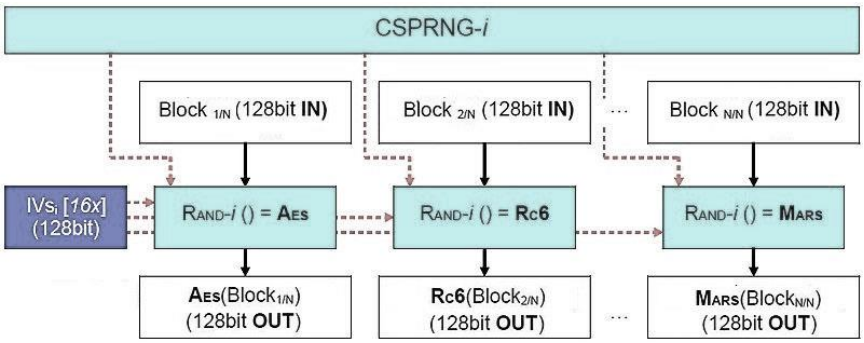
- Mỗi mật mã có một thiết lập độc lập:

$Cipher_j$ = { IV_j, Key_j }

- Mỗi khối dữ liệu được xử lý với một mật mã khác nhau, được chọn bằng cách sử dụng CSPRNG:

$CryptedBlock_k$ ={ r ← Rand-i (); $Cipher_r$ (IV_r, Key_r, $Block_k$)}

Hình sau thể hiện quá trình chọn mật mã sử dụng CSPRNG:



Hình 2.23. Sử dụng CSPRNG để chọn thuật toán mã hóa ngẫu nhiên.

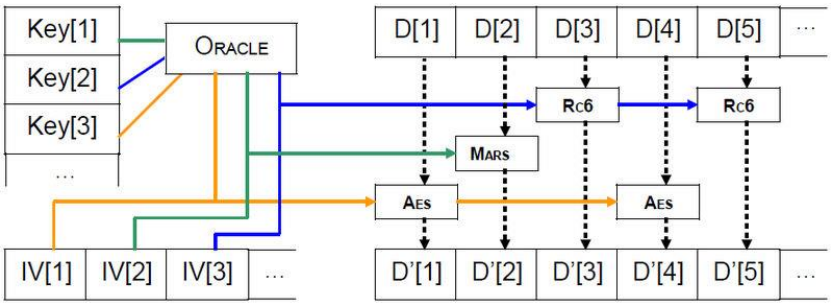
- Quá trình thiết lập Cryptography và thiết lập CSPRNG sử dụng 2 mật khẩu độc lập với nhau.

- Mỗi mật mã có một IV và một khóa khác nhau.

- Mảng IV được mã hóa sử dụng các thuật toán mã hóa chọn ngẫu nhiên, với công thức mã hóa như sau:

$CryptedIVs_n$ = Crypt (IVs_n, $CryptedIVs_{n-1}$).

- CSPRNG làm việc giống như một hệ thống Random Oracle, nó cung cấp công cụ mã hóa trong suốt cả quá trình lựa chọn của mình, một khóa sẽ được kết hợp với một mật mã nào đó và mật mã đó sẽ áp dụng cho các khối dữ liệu được chọn,… [6]



Hình 2.24. Mô hình làm việc của một CSPRNG trong Multi-Cryptography.

2.4. Ưu điểm và nhược điểm của mô hình bảo mật 4 lớp

2.4.1. Ưu điểm của mô hình bảo mật 4 lớp

Mô hình bảo mật 4 lớp có các ưu điểm sau:

- Sử dụng nhiều lớp để ẩn và bảo vệ dữ liệu:

+ Sử dụng mã hóa đối xứng 256 bit + 256 bit (với phần mở rộng mật khẩu KDF4), với nhiều thuật toán mã hóa hiện đại để mã hóa dữ liệu (16 thuật toán).

+ Dữ liệu được Crambling kết hợp với khóa đối xứng 256 bit (xáo trộn dựa trên CSPRNG).

+ Dữ liệu được Whitening cùng với khóa đối xứng 256 bit (trộn nhiễu dựa trên CSPRNG).

+ Mã hóa các bit của tập tin vận chuyển bằng mã hóa phi tuyến tính.

- Sử dụng nhiều mật khẩu trong quá trình mã hóa và che dấu thông tin (3 mật khẩu).

- Có thể che dấu thông điệp trong các tập tin vận chuyển có kích thước rất lớn.

- Thông điệp bí mật được phân chia thành nhiều khối và có thể ẩn trong nhiều tập tin vận chuyển.

- Cung cấp chức năng Deniable Steganography, có thể tạo thêm nhiều dữ liệu mồi Decoy ít quan trọng, giúp dữ liệu bí mật được an toàn hơn.

2.4.2. Nhược điểm của mô hình bảo mật 4 lớp

Theo nguyên văn của tác giả, các chuyên gia phía EmbeddedSW Company thì mô hình bảo mật 4 lớp này "không có nhược điểm".

Lý do này được giải thích như sau: Steganalysis và Steganography cũng giống như trò chơi ẩn giấu và tìm kiếm. Một khi kẻ tấn công thông minh hơn thì chúng ta phải cải thiện kỹ thuật ẩn giấu của mình. Không có cách nào tuyệt đối để đo chất lượng của Steganography. Chúng ta chỉ có thể đo mức độ mạnh mẽ của Steganography so với các kỹ thuật tấn công hiện tại.

Nói một cách chủ quan thì cách tiếp cận bảo mật 4 lớp hiện nay là mạnh mẽ nhất. Các phương pháp tiếp cận khác đang được mô tả theo lý thuyết nhưng chúng chưa hoàn thiện và ứng dụng trong thực tế.

2.5. Kết luận chương 2

Chương 2 đã hoàn thành việc nghiên cứu về mô hình bảo mật 4 lớp và các thuật toán liên quan, cụ thể như sau:

- Tổng quan về mô hình bảo mật 4 lớp: Phần này tập trung giới thiệu về mô hình bảo mật 4 lớp và các lớp trong mô hình, kiến trúc tổng quan của mô hình.

- Chức năng của các lớp trong mô hình bảo mật 4 lớp: Trình bày kiến trúc, chức năng và cách thức hoạt động của từng lớp trong mô hình.

- Các thuật toán được sử dụng trong mô hình bảo mật 4 lớp: Giới thiệu các thuật toán được sử dụng trong mô hình và trình bày 2 thuật toán tiêu biểu trong mô hình là: Thuật toán Cryptographically Secure Pseudo-Random Number Generator dựa trên AES - CTR và thuật toán Multi-Cryptography - 256 + 256 bit - CBC - Segment.

- Ưu điểm và nhược điểm của mô hình bảo mật 4 lớp: Trình bày các ưu nhược điểm của mô hình bảo mật 4 lớp.

Tóm lại, ở chương này đã giới thiệu tổng quan về kiến trúc mô hình bảo mật 4 lớp, tập trung trình bày lý thuyết cơ bản về chức năng từng lớp, các thuật toán liên quan được sử dụng trong mô hình. Ngoài ra trong phần mềm bày về ưu và nhược điểm của mô hình bảo mật 4 lớp nhằm đánh giá một cách khách quan nhất về nó.

Chương 3
PHẦN MỀM THỬ NGHIỆM ÁP DỤNG MÔ HÌNH BẢO MẬT 4 LỚP VÀO GIẤU TIN TRONG DỮ LIỆU ĐA PHƯƠNG TIỆN

Trong chương 3, nội dung đề cập đến việc giới thiệu về phần mềm OpenPuff. Việc áp dung mô hình bảo mật 4 lớp trong phần mềm này để thực hiện quá trình giấu tin trong các dữ liệu đa phương tiện. Phần này còn mô tả chi tiết các bước để sử dụng các chức năng trong phần mềm. Tập trung chủ yếu vào 2 chức năng chính là ẩn và giải ẩn dữ liệu bí mật được giấu trong các tập tin dữ liệu đa phương tiện. Giúp chúng ta có cái nhìn rõ nét về phần mềm cũng như chức năng của mô hình bảo mật 4 lớp được ứng dụng trong thực tiễn.

3.1. Giới thiệu về phần mềm OpenPuff

OpenPuff tên gọi đầy đủ là OpenPuff Steganography and Watermarking, tên gọi tắt là OpenPuff hoặc Puff, là một công cụ Steganography bán mã nguồn mở, miễn phí dành cho Microsoft Windows.

OpenPuff được sáng tạo bởi Cosimo Oliboni của EmbeddedSW Company và được duy trì như một phần mềm độc lập bằng cách ứng dụng mô hình bảo mật 4 lớp.

Phần mềm này là công cụ Steganography đầu tiên sử dụng mô hình bảo mật 4 lớp. Phiên bản đầu tiên 1.01 phát hành vào tháng 12 năm 2004. Phiên bản hiện tại là 4.0 được phát hành 7/7/2012.

Các nền tảng hỗ trợ phần mềm OpenPuff là:

- Windows 7 (32bit/64 bit).

- Windows Vista (32bit/64bit).

- Windows XP.

OpenPuff hỗ trợ nhiều định dạng cho các tập tin vận chuyển:

- Image: BMP, JPG, PCX, PNG, TGA).

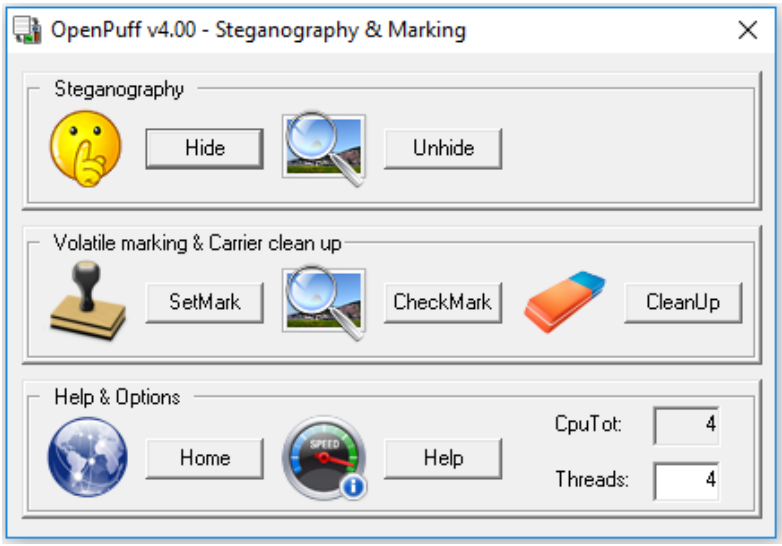
- Audio: AIFF, MP3, NEXT/SUN, WAV.

- Video: 3GP, MP4, MPG, VOB.

- Flash-Adobe: FLV, SWF, PDF.

Ta có thể tải miễn phí phần mềm OpenPuff trên trang web Embeddedsw.net.

Dưới đây là giao diện chính của phần mềm:



Hình 3.1. Giao diện chính của phần mềm OpenPuff.

OpenPuff Steganography là một dự án bán mã nguồn mở trong đó thuật toán mã hóa là mã nguồn mở, nhưng phần còn lại của phần mềm là độc quyền. Cách OpenPuff hoạt động là dữ liệu được ẩn được chia nhỏ và ẩn bên trong các tệp tin vận chuyển bằng nhiều cách nhúng.

Phần mềm cho phép người dùng ẩn dữ liệu trong nhiều loại tập tin vận chuyển khác nhau như các tập tin hình ảnh, âm thanh và video. Trước khi dữ liệu được ẩn, nó được Encrypted, Scrambled, Whitened, sau đó Encoded bằng cách sử dụng mô hình bảo mật 4 lớp.

OpenPuff tập trung vào việc bảo mật những thông tin ẩn. Các lớp bảo mật của OpenPuff ẩn đi những thông tin bằng cách mã hóa dữ liệu và bảo vệ nó bằng 3 lớp mật khẩu khác nhau. Mỗi mật khẩu phải có ít nhất 8 kí tự để yêu cầu giấu dữ liệu bằng OpenPuff.

Dữ liệu có thể được dấu thông qua nhiều tập tin vận chuyển, cho phép 1 lượng lớn thông tin bí mật được giấu đi. OpenPuff còn có thể bảo vệ những thông tin ẩn bằng cách thêm 1 lượng lớn dữ liệu nhiễu ngẫu nhiên vào thông tin trước khi nó được mã hóa và giấu đi.

Một điểm đặc biệt của OpenPuff là Deniable Steganography, nó cho phép 2 phần riêng rẽ của dữ liệu được ẩn trong những tệp đính kèm. Điều này cho phép ta có thể ẩn đi cả thông tin nhạy cảm và thông tin mồi. Nếu bắt buộc phải tiết lộ mật khẩu bảo vệ dữ liệu đã được ẩn, 1 người sử dụng có thể bỏ đi mật khẩu mồi, tức là tiết lộ thông tin mồi, thông tin không bí mật, trong khi

những thông tin bí mật vẫn được giữ ở trạng thái ẩn và được bảo vệ bởi 1 phần của mật khẩu bí mật.

OpenPuff còn cho phép thêm những chuỗi ẩn có độ dài lên đến 32 kí tự vào một tệp đính kèm, nhằm mục đích đánh dấu quyền sở hữu tập tin. Đây là 1 kiểu Digital Watermark. Digital Watermark có thể bị tiết lộ mà không cần mật khẩu, sử dụng chức năng CheckMark ở trong OpenPuff. Chức năng Digital Watermark có thể được sử dụng để nhận dạng hoặc theo vết những tệp tin được đăng tải công khai hoặc được chia sẻ trên Internet [5].

OpenPuff là một công cụ quan trọng cho các nhà nghiên cứu bảo mật và cho bất cứ ai cần chia sẻ thông điệp bí mật. Phần tiếp theo sẽ trình bày về các chức năng của phần mềm này.

3.2. Các chức năng của phần mềm OpenPuff

Phần mềm OpenPuff là một công cụ Steganography & Marking với 3 nhóm chức năng:

- Steganography.

- Volatile marking & Carrier clean up.

- Help & Option.

3.2.1. Nhóm chức năng Steganography của phần mềm OpenPuff

Steganography là nhóm chức năng chính của phần mềm, được xây dựng dựa trên mô hình bảo mật 4 lớp. Ở chức năng này chúng ta có hai sự lựa chọn là Hide và Unhide dữ liệu.

Trong đó, Hide là chức năng ẩn một thông điệp bí mật vào các tập tin vận chuyển với mô hình bảo mật 4 lớp.

Quá trình ẩn dữ liệu được thực hiện qua 4 bước gồm:

- (1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32).

- (2) Data (Max: 256Mb).

- (3) Carrier selection (Order Sensitive).

- (4) Bit selection options.

Ngoài ra còn có thêm phần tùy chọn “Add Decoy!”, một chức năng Deniable Steganography.

Về chi tiết các bước sẽ được trình bày rõ hơn trong phần thử nghiệm.

Chức năng còn lại trong nhóm chức năng này là Unhide. Một chức năng ngược lại với chức năng Hide, dùng để giải ẩn một thông điệp bí mật hoặc thông điệp Decoy từ các tập tin vận chuyển với mô hình bảo mật 4 lớp.

Quá trình giải ẩn chỉ với 3 bước sau:

- (1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32).

- (2) Carrier selection (Order Sensitive).

- (3) Bit selection options.

Một lưu ý trong sử dụng chức năng này là mọi thông tin về quá trình giải ẩn phải hoàn toàn giống với quá trình ẩn thì mới có thể giải ẩn thành công. Về chi tiết của chức năng này cũng sẽ được trình bày chi tiết ở phần thử nghiệm.

3.2.2. Nhóm chức năng Volatile marking & Carrier clean up của phần mềm OpenPuff

Đây là nhóm chức năng được dùng với mục đích tạo và kiểm tra thông tin bản quyền.

Trong đó:

- Chức năng SetMark dùng để đánh đấu thông tin mà chúng ta muốn đánh dấu vào dữ liệu.

- Chức năng CheckMark dùng để kiểm tra thông tin đã được đánh dấu trong thông điệp.

- Chức năng CleanUp dùng để xóa thông tin đã được đánh dấu trước đó trong thông điệp.

3.2.3. Nhóm chức năng Help & Option của phần mềm OpenPuff

Đây là phần chức năng giúp đỡ người sử dụng về phần mềm và hỗ trợ tốc độ xử lý của phần mềm.

Trong đó:

- Home và Help giúp chúng ta đến truy cập đến các tài liệu trên trang chủ của phần mềm.

- Phần còn lại là Cpu Total và Thread, giúp chúng ta có thể tùy chỉnh số Thread máy tính cho phép phần mềm sử dụng, số Thread tối đa phụ thuộc vào cấu hình của máy tình.

3.3. Bài toán thử nghiệm

Kịch bản thử nghiệm được đưa ra là: Hai đại sứ quán của nước ta ở hai nước khác nhau, tiến hành liên lạc trao đổi các thông điệp bí mật sử dụng phần mềm OpenPuff.

Trước hết, hai đại sứ quán sẽ quy ước sử dụng chung các mật khẩu giấu tin. Khi đại sứ quán A muốn truyền thông điệp bí mật cho đại sứ quán B, thông điệp sẽ được ẩn giấu vào các tệp dữ liệu đa phương tiện nhờ sử dụng phần mềm OpenPuff. Sau đó tệp dữ liệu đa phương tiện được họ đăng tải trên trang Facebook cá nhân như một bài viết thông thường. Đại sứ quán B lên trang Facebook của đại sứ quan A tải tệp tin đó về và tiến hành giải ẩn thông điệp bí mật mà đại sứ quán A muốn gửi. Quá trình truyền gửi tin được tiến hành bí mật và đảm bảo an toàn trong suốt quá trình truyền nhận. Ngoài ra các tệp dữ liệu đa phương tiện được các đại sứ quán tự tạo ra bằng cách tự chụp, tự quay,… đảm bảo chưa từng được phân phối trên mạng nhằm tăng tính bảo mật cho quá trình truyền nhận.

Cụ thể việc ẩn và giải ẩn dữ liệu bằng phần mềm OpenPuff sẽ được trình bày chi tiết ở các phần dưới đây.

3.3.1. Ẩn dữ liệu bằng phần mềm OpenPuff

Để tiến hành quá trình ẩn dữ liệu bằng phần mềm OpenPuff, đầu tiên ta chọn nút “Hide” trong chức năng Steganography ở màn hình chính của phần mềm OpenPuff để mở giao điện chức năng ẩn dữ liệu.

Sau đó ta tiến hành các bước ẩn dữ liệu như sau:

- *Bước 1:* Nhập 3 mật khẩu không tương quan với nhau ở phần (1). Trong đó:

+ Cryptography (A): Là mật khẩu thứ nhất (Cryptography Keys).

+ Cryptography (B): Là mật khẩu thứ hai (Cryptography CSPRNG).

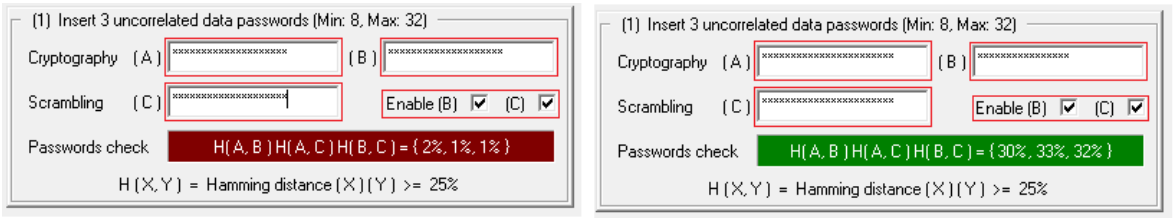
+ Scrambling (C): Là mật khẩu thứ ba (Scrambling CSPRNG).

+ Enable (B): Enable/Disable mật khẩu thứ hai.

+ Enable (C): Enable/Disable mật khẩu thứ ba.

Ta tiến hành nhập 3 mật khẩu riêng biệt. Mỗi mật khẩu phải khác nhau ở mức độ bit và có chiều dài tối thiểu là 8, tối đa là 32 ký tự. Phần nhập mật khẩu có thể dễ dàng tùy chỉnh vô hiệu hóa mật khẩu thứ hai (B) và/hoặc mật

khẩu thứ ba (C). Những mật khẩu bị vô hiếu hóa được đặt như mật khẩu thứ nhất (A).



Hình 3.2. Nhập mật khẩu chưa hợp lệ và hợp lệ.

Ví dụ: Ta nhập các mật khẩu “DataPssw1” (A) “DataPssw2” (B) “DataPssw3” (C):

```
(A) 01000100 01100001 01110100 01100001 01010000 01110011 01110011 01110111 00110001
(B) 01000100 01100001 01110100 01100001 01010000 01110011 01110011 01110111 00110010
(C) 01000100 01100001 01110100 01100001 01010000 01110011 01110011 01110111 00110011
```

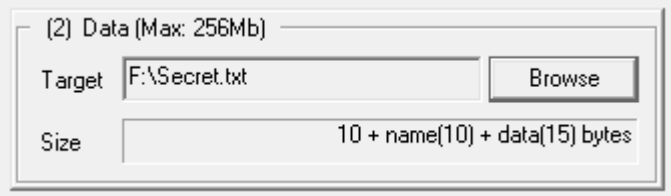
Ta có: (A ∩ B) 98%, (A ∩ C) 99%, (B ∩ C) 99%. Suy ra Haming Distance < 25%, không thỏa mãn điều kiện.

Một ví dụ khác: Ta nhập các mật khẩu “DataPssw1” (A) “DataPssw2” (B) “DataPssw3” (C):

```
(A) 01000110 01101001 01110010 01110011 01110100 01000100 01100001 01110100 01100001
(B) 01010011 01100101 01100011 01101111 01101110 01100100 01000100 01100001 01110100
(C) 01000001 01101110 01101111 01110100 01101000 01100101 01110010 01000100 01100001
```

Ta có: (A ∩ B) 70%, (A ∩ C) 67%, (B ∩ C) 68%. Suy ra Haming Distance >= 25%, thỏa mãn điều kiện.

- *Bước 2:* Chọn dữ liệu bí mật mà chúng ta muốn ẩn (thường là các tập tin lưu trữ zip/rar…) với kích thước tối đa của tập tin dữ liệu bí mật là 256 Mb.



Hình 3.3. Chọn tập tin chứa dữ liệu bí mật.

- *Bước 3:* Tiến hành lựa chọn các tập tin vận chuyển ở phần (3). Trong đó:

+ Shuffle: Xáo trộn ngẫu nhiên thứ tự tất cả các tập tin vận chuyển.

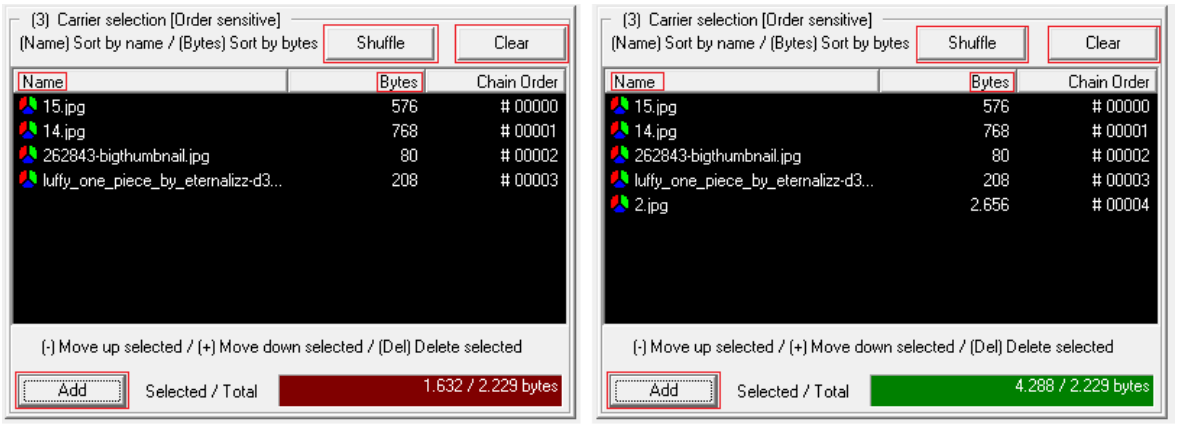
+ Clear: Hủy tất cả các tập tin vận chuyển đã chọn.

+ Add: Thêm một tập tin vận chuyển mới vào danh sách.

+ Name/Bytes: Sắp xếp tất cả các tập tin vận chuyển theo tên hoặc theo byte.

+ (+)/(-): Di chuyển tập tin được chon lên hoặc xuống theo thứ tự.

+ Del: Xóa tập tin vận chuyển đã chọn.



Hình 3.4. Chọn tập tin vận chuyển thỏa mã và không thỏa mãn.

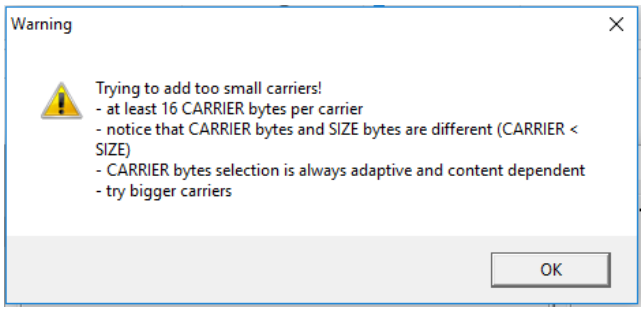
Trong khi Selected bytes < Total bytes ta phải:

+ Thêm một tập tin vận chuyển mới.

+ Tăng mức lựa chọn bit.

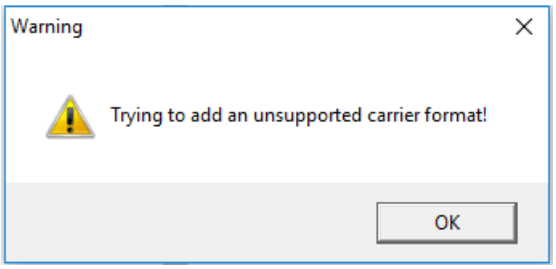
Một số tập tin vận chuyển sẽ không được thêm vào vì các ràng buộc trong quá trình Steganography:

+ Cảnh báo các tập tin vận chuyển có kích thước chưa đủ.



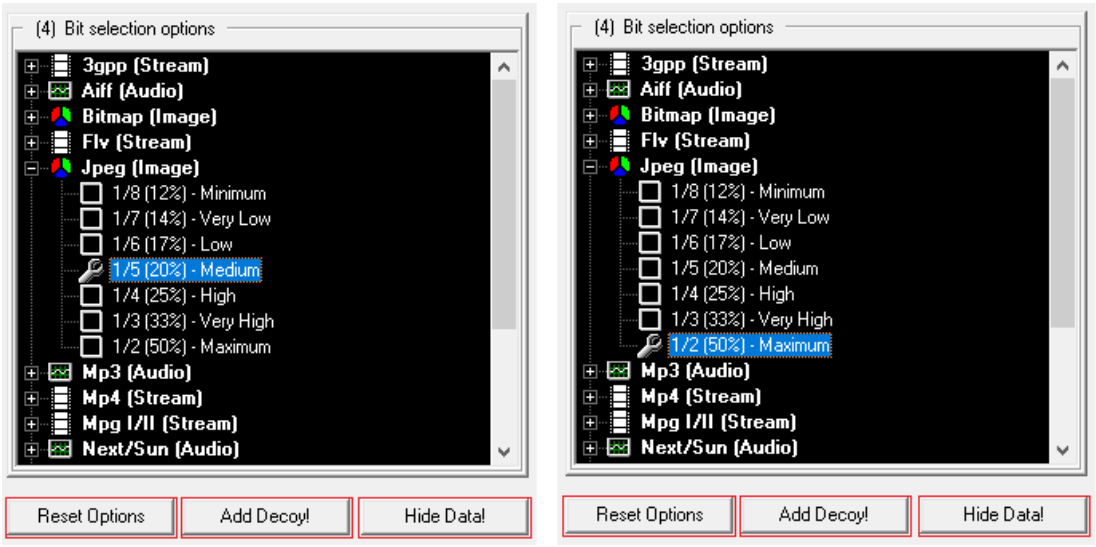
Hình 3.5. Cảnh báo các tập tin vận chuyển có kích thước chưa đủ.

+ Cảnh báo tập tin vận chuyển có định dạng không được hỗ trợ.



Hình 3.6. Cảnh báo tập tin vận chuyển có định dạng không được hỗ trợ.

- *Bước 4:* Tùy chọn bit selection.



Hình 3.7. Tùy chọn bit selection.

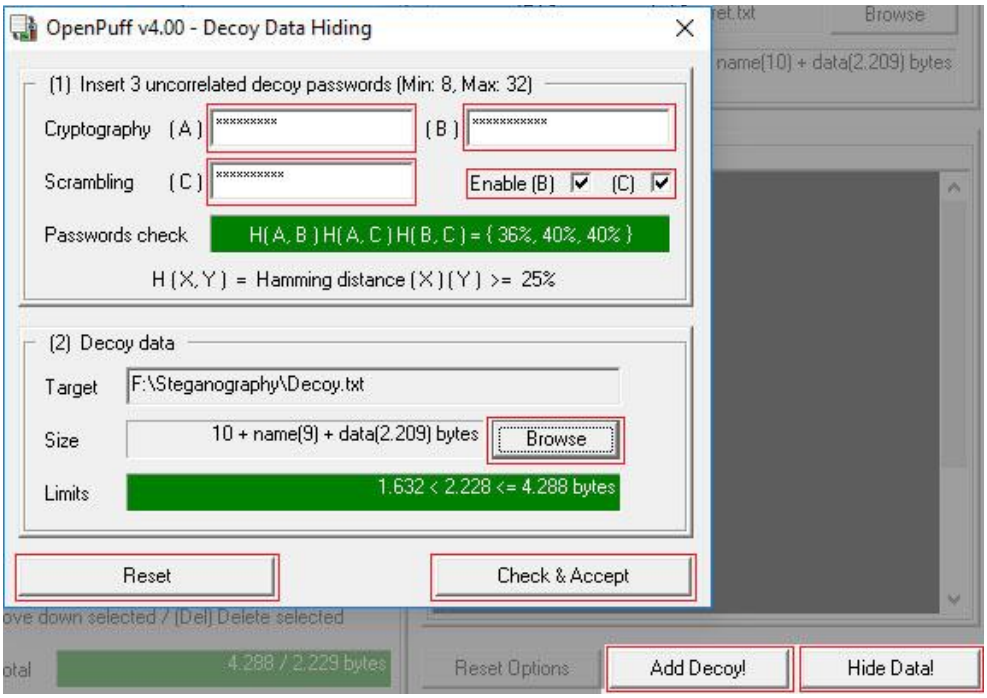
Trong đó:

+ Reset Options: Reset tất cả các mức bit selection trở lại mặc định.

+ Add Decoy!: Thêm một Decoy (Deniable Steganography)

+ Hide Data!: Bắt đầu quá trình ẩn dữ liệu.

Tùy chọn không bắt buộc “Add Decoy!”:



Hình 3.8. Decoy Data Hiding.

Trong đó:

+ Cryptography A: Là mật khẩu thứ nhất (Cryptography Keys).

+ Cryptography B: Là mật khẩu thứ hai (Cryptography CSPRNG).

+ Scrambling C: Là mật khẩu thứ ba (Scrambling CSPRNG).

+ Enable B: Enable/Disable mật khẩu thứ hai.

+ Enable C: Enable/Disable mật khẩu thứ ba.

+ Browse: Chọn một tập tin.

+ Reset: Reset mật khẩu và tập tin.

+ Check & Accept: Kiểm tra tương quan mật khẩu và kích thước tập tin, sau đó tiến hành ẩn dữ liệu Decoy.

Chúng ta có thể thêm mật khẩu Decoy và dữ liệu Decoy:

+ Các mật khẩu Decoy phải khác nhau và khác với mật khẩu của dữ liệu ẩn.

+ Các mật khẩu Decoy cũng có thể được tùy chỉnh như chúng ta tùy chỉnh với mật khẩu của dữ liệu ẩn.

+ Dữ liệu Decoy phải tương thích với dữ liệu nhạy cảm.

$\sum k \in \{1, N-1\}$ used_carrier_bytes ($carr_k$) < Sizeof (Decoy) ≤ $\sum k \in \{1, N\}$ used_carrier_ bytes ($carr_k$)

Sau khi thực hiện các bước:

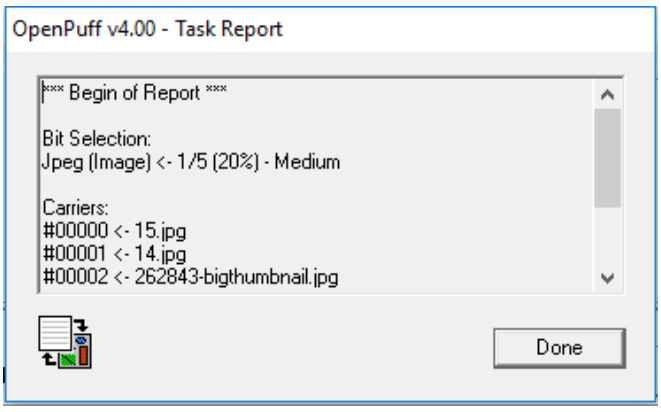
+ Nhập các mật khẩu, ít nhất 8 ký tự.

+ Thêm những Carrier bit đủ lớn.

+ Thêm một Decoy (không bắt buộc).

Chúng ta chọn "Hide Data!" để bắt đầu tiến trình ẩn dữ liệu.

Sau khi quá trình ẩn dữ liệu thành công sẽ xuất hiện một báo cáo tóm tắt tất cả thông tin cần thiết về quá trình ẩn dữ liệu [4].



Hình 3.9. Thông báo tóm tắt quá trình ẩn dữ liệu.

3.3.2. Giải ẩn dữ liệu bằng phần mềm OpenPuff

Để giải ẩn dữ liệu đầu tiên ta chọn nút "Unhide" trong chức năng Steganography ở màn hình chính của phần mềm OpenPuff.

Giao điện "Data Unhiding" xuất hiện. Chúng ta bắt đầu thực hiện các bước giải ẩn dữ liệu như sau:

- *Bước 1:* Thêm các mật khẩu của chúng ta vào phần (1) của giao điện (mật khẩu bí mật để lấy dữ liệu bí mật, mật khẩu Decoy để lấy dữ liệu Decoy).

Trong đó:

+ Cryptography (A): Là mật khẩu thứ nhất (Cryptography Keys).

+ Cryptography (B): Là mật khẩu thứ hai (Cryptography CSPRNG).

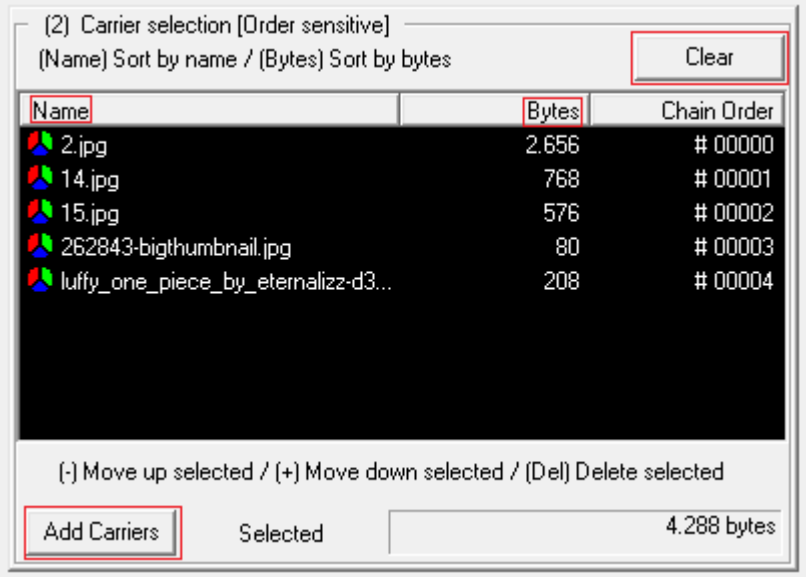
+ Scrambling (C): Là mật khẩu thứ ba (Scrambling CSPRNG).

+ Enable (B): Enable/Disable mật khẩu thứ hai.

+ Enable (C): Enable/Disable mật khẩu thứ ba.

- *Bước 2:* Thêm tất cả các tập tin vận chuyển đã được xử lý trong suốt quá trình ẩn dữ liệu ở phần (2) của giao diện.

Chú ý một điều là các tập tin vận chuyển phải được sắp xếp theo đúng thứ tự như khi thực hiện quá trình ẩn dữ liệu.



Hình 3.10. Các tập tin vận chuyển được sử dụng để giải ẩn dữ liệu.

Trong đó:

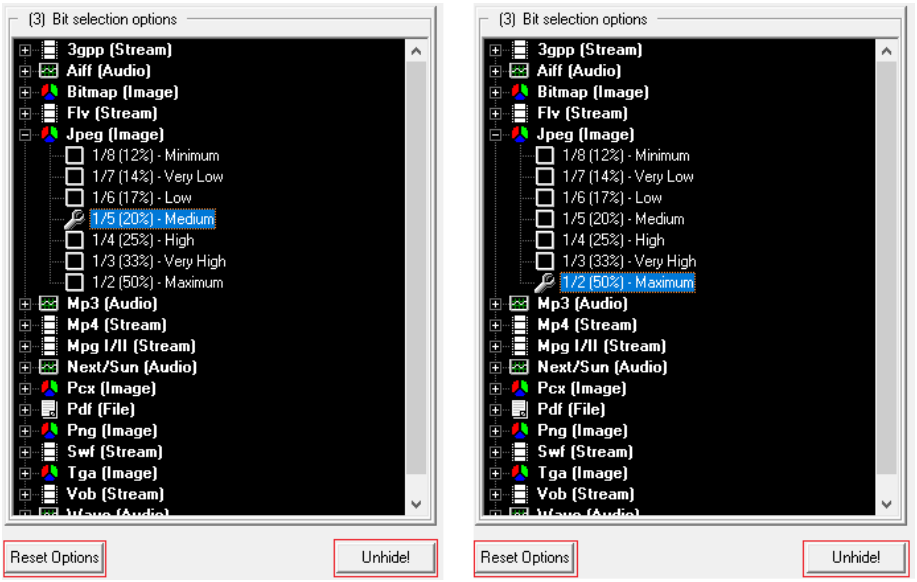
+ Clear: Hủy tất cả các tập tin vận chuyển đã chọn.

+ Add Carriers: Thêm các tập tin vận chuyển mới vào danh sách.

+ Name/Bytes: Sắp xếp tất cả các tập tin vận chuyển theo tên hoặc theo byte.

+ (+)/(-): Di chuyển tập tin được chon lên hoặc xuống theo thứ tự.

- *Bước 3:* Tùy chọn bit option.



Hình 3.11. Tùy chọn bit Option.

Trong đó:

+ Reset Options: Reset tất cả các mức bit Selection.

+ Unhide!: Bắt đầu tiến trình giải ẩn.

Ở bước này ta tiến hành thiết đặt các giá trị bit Selection cho đúng với giá trị ban đầu của nó khi thực hiện ẩn dữ liệu.

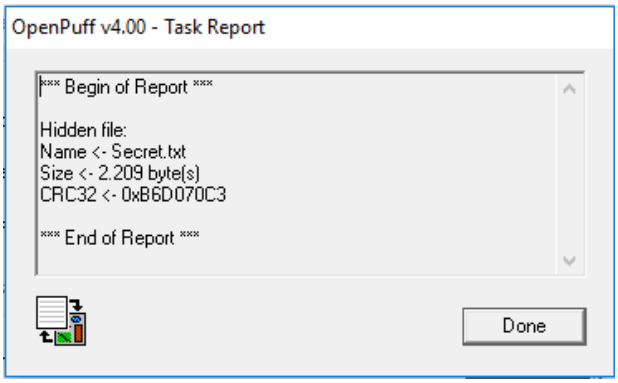
Sau khi tiến hành xong các bước:

+ Nhập mật khẩu.

+ Thêm tất cả các tập tin vận chuyển.

+ Thiết đặt các giá trị bit Selection như giá trị ban đầu khi ẩn dữ liệu.

Ta sẽ chon nút “Unhide!” để bắt đầu tiến trình mở dữ liệu ẩn hoặc dữ liệu Decoy.



Hình 3.12. Thông báo tóm tắt quá trình giải ẩn.

Nếu các tập tin vận chuyển đã được thêm theo thứ tự đúng, với các mức lựa chọn bit ban đầu, OpenPuff sẽ có thể tái tạo lại dữ liệu ban đầu. Để bảo mật tốt hơn, dữ liệu sẽ được tái tạo lại chỉ sau khi kiểm tra dư thừa tuần hoàn CRC thành công [4].

Chú ý rằng: Ngay cả những thay đổi nhỏ nhất của một trong những tập tin vận chuyển có thể làm hỏng dữ liệu và ngăn chặn mọi cố gắng giải ẩn.

3.4. Khả năng ứng dụng của đề tài trong công tác Công an

Kết quả nghiên cứu của đề tài có thể được ứng dụng trong công tác truyền thông, liên lạc nội bộ trong Ngành Công an.

Việc ứng dụng mô hình bảo mật 4 lớp cũng như phần mềm OpenPuff có thể được áp dụng vào việc truyền thông, liên lạc bí mật trong các công tác Ngành Công an như: Tình báo, trinh sát, sử dụng đặc tình, các quá trình trao đổi các thông tin quan trọng,… thay thế cho các phương thức liên lạc truyền thống khác như: Sử dụng hộp thư sống, hộp thư chết, các phương pháp mã hóa, che giấu thông tin thông thường.

Với phương thức liên lạc mới này có thể đảm bảo tốc độ truyền gửi thông tin cao mà vẫn đảm bảo được sự an toàn và bí mật cho thông tin.

Ví dụ: Các nhân viên tình báo của Tổng cục Tình báo Công an có thể trao đổi hoặc báo cáo thông tin với chỉ huy bằng cách sử dụng phần mềm OpenPuff để ẩn các thông điệp vào các tập tin đa phương tiện mà họ tự tạo ra. Các tập tin đa phương tiện có thể là các bức ảnh tự chụp, các đoạn video tự quay,... Sau đó chúng được sử dụng để ẩn thông điệp bí mật nhờ sử dụng phần mềm OpenPuff với các mật khẩu được thỏa thuận trước. Các tập tin sau có thể được gửi đi qua đường Internet mà vẫn đảm bảo được độ bảo mật cao trong quá trình trao đổi. Bên nhận cũng sử dụng phần mềm OpenPuff với các mật khẩu đã thỏa thuận trước đó để giải ẩn thông điệp bí mật.

3.5. Kết luận chương 3

Chương 3 đã hoàn thành việc trình bày các nội dung sau:

- Giới thiệu về phần mềm OpenPuff: Giới thiệu tổng quan về phần mềm OpenPuff.

- Các chức năng của phần mềm OpenPuff: Trình bày các chức năng của phần mềm OpenPuff như: Steganography, Volatile marking & Carrier clean up, Help & Option.

- Áp dụng mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện nhờ phần mềm OpenPuff: Trình bày quá trình ẩn và giải ẩn thông điệp bí mật bằng phần mềm OpenPuff.

- Khả năng ứng dụng của đề tài trong công tác Công an: Trình bày các lĩnh vực trong Ngành Công an có thể áp dụng mô hình bảo mật 4 lớp cũng như phần mềm OpenPuff vào trong công tác thực tiễn của cán bộ, chiến sĩ.

Tóm lại, ở chương 3 đã giải quyết vấn đề áp dụng mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện bằng cách sử dụng phần mềm OpenPuff. Chương này đã trình bày khái quát nhất về việc mô hình bảo mật 4 lớp được áp dụng trong OpenPuff để tiến hành ẩn và giải ẩn các thông điệp bí mật được giấu trong các tập tin đa phương tiện. Làm nổi bật các chức năng được thiết kế trong phần mềm OpenPuff như: Chức năng Steganography nhờ sử dụng mô hình bảo mật 4 lớp, chức năng đánh dấu quyền sở hữu của tập tin bằng cách sử dụng Digital Watermark. Ngoài ra, ở chương này còn trình bày về khả năng ứng dụng của đề tài trong công tác Công an nhằm cải thiện và nâng cao hiểu quả trong công tác của các cán bộ chiến sĩ phục vụ trong Ngành.

KẾT LUẬN

1. Các kết quả nghiên cứu của ĐATN

Sau một thời gian nghiên cứu tích cực, nghiêm túc, tôi đã hoàn thành Đồ án tốt nghiệp với đề tài "*Nghiên cứu mô hình bảo mật 4 lớp, áp dụng vào giấu tin trong dữ liệu đa phương tiện*", đảm bảo yêu cầu về nội dung, chất lượng và tiến độ đề ra. Cụ thể:

- **Những điểm đạt được:**

- Đã tìm hiểu, nghiên cứu và trình bày được tổng quan về lĩnh vực Steganography.

- Nghiên cứu, tìm hiểu, trình bày lý thuyết cơ bản về mô hình bảo mật 4 lớp và kiến trúc của nó, các chức năng của từng lớp trong mô hình. Trình bày 2 thuật toán tiêu biểu được sử dụng trong mô hình bảo mật 4 lớp này.

- Nghiên cứu về việc áp dụng mô hình bảo mật 4 lớp vào giấu tin trong dữ liệu đa phương tiện bằng phần mềm bán mã nguồn mở OpenPuff, làm nổi bật từng chức năng của phần mềm.

- Báo cáo được thực hiện đúng nội dung, tiến độ, được bố cục, trình bày theo đúng chuẩn quy định. Trình bày phần thử nghiệm rõ ràng, chi tiết.

- **Những điểm còn tồn tại:**

Do trình độ, khả năng và thời gian còn hạn chế nên báo cáo vẫn còn một số tồn tại như sau:

- Mặc dù đã tích cực, chủ động tìm tòi các nguồn tài liệu, báo cáo hội thảo nước ngoài bằng Tiếng Anh để nghiên cứu, nhưng việc thể hiện lại trên ngôn ngữ Tiếng Việt còn gặp bối rối, một số thuật ngữ chuyên ngành chưa được truyền tải chính xác.

- Trong khuôn khổ báo cáo chưa thể trình bày chi tiết các kiến thức, đi sâu vào các thuật toán trong các lớp của mô hình bảo mật 4 lớp do vấn đề độc quyền của một số phần trong mô hình.

- Nguồn các tài liệu tham khảo không nhiều, hầu như không có tài liệu tham khảo bằng Tiếng Việt, tài liệu Tiếng Anh thì rất ít đặc biết là các tài liệu chuyên sâu.

2. Hướng phát triển của ĐATN

Đề tài này có thế được phát triển theo một số hướng như sau:

- Tiếp tục nghiên cứu chuyên sâu về lĩnh vực Steganography trong an toàn thông tin.

- Tiếp tục nghiên cứu về mô hình bảo mật 4 lớp, làm rõ từng thuật toán được sử dụng trong mô hình.

- Tiếp tục nghiên cứu về tách tin từ tập tin đa phương tiện và quá trình giải mã tin.

- Cải tiến các thuật toán được sử dụng trong mô hình bảo mật 4 lớp để tăng khả năng bảo mật.

- Xây dựng một phần mềm tự thiết kế dựa trên mô hình bảo mật 4 lớp để thực hiện việc giấu tin trong dữ liệu đa phương tiện, tương tự phần mềm OpenPuff dựa trên bộ lõi mã nguồn mở của nó.

TÀI LIỆU THAM KHẢO

1. Tiếng Việt

[1] Nguyễn Thanh Cường (2009), "Giấu tin trong ảnh và ứng dụng trong an toàn bảo mật thông tin", *Khóa luận tốt nghiệp, Trường Đại Học Công Nghệ, Đại Học Quốc Gia Hà Nội*, tr. 6-9.

[2] Nguyễn Diễm Hương (2012), "Kỹ thuật giấu tin trên k bit LSB của ảnh", *Đồ án tốt nghiệp, Trường Đại Học Dân Lập Hải Phòng*, tr.15-16.

2. Tiếng Anh

[3] Yao Lu (2014), "Investigating Steganography in Audio Stream for Network Forensic Investigations: Detection and Extraction", *Auckland, New Zealand*, pp. 6-19, 71-73.

[4] Cosimo Oliboni (2011), "OpenPuff Help: Steganography & Watermarking", *Italy*.

[5] Michael Chesbro (2014), "OpenPuff: Steganography & Watermarking Tool".

[6] Cosimo Oliboni (2011), "Multiobfuscator: Cryptography & Obfuscation", *Italy*, pp. 3-13.

3. Website

[7] Https://en.wikipedia.org/wiki/Steganography.

[8] Https://en.wikipedia.org/wiki/Steganography_tools.

[9] Https://en.wikipedia.org/wiki/Obfuscation.

[10] Https://en.wikipedia.org/wiki/Digital_watermarking.

[11] Https://vi.wikipedia.org/wiki/AES.

[12] Https://en.wikipedia.org/wiki/Block_cipher_Mode_of_operation.

[13] Http://embeddedsw.net/libObfuscate_Cryptography_Home.