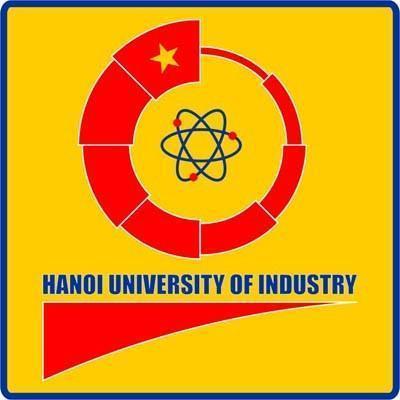
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

∙∙∙🙞🙜🕮🙞🙜∙∙∙

****

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM**

**Học phần: An toàn và bảo mật thông tin**

**ĐỀ TÀI 8:**

**ỨNG DỤNG HỆ THỐNG MÃ HOA LAI VÀO CÔNG TÁC BẢO MẬT TRONG TRUYỀN TẢI ĐỀ THI**

**GVHD:TS. Nguyễn Bá Nghiễn**

**Nhóm : 14**

Thành viên nhóm:

1. Nguyễn Tiến Duy - 2020604925
2. Nguyễn Tiến Kiên –2021608239
3. Trần Đoàn Xuân Thành–2021607834
4. Phạm Quốc Tiến – 2021608512
5. Đào Văn Nhật – 20220202020

**Hà Nội – 2023**

**MỤC LỤC**

[Chương 1. Tổng quan 6](#_Toc138104867)

[1.1. Tổng quan về an toàn và bảo mật thông tin 6](#_Toc138104868)

[1.1.1.Sự cần thiết của việc đảm bảo an toàn thông tin 6](#_Toc138104869)

[1.1.2.Khái niệm an toàn thông tin 6](#_Toc138104870)

[1.1.3.Các yêu cầu an toàn bảo mật thông tin 6](#_Toc138104871)

[1.1.4.Các phương pháp bảo vệ thông tin 8](#_Toc138104872)

[1.2. An toàn thông tin bằng mật mã 8](#_Toc138104873)

[1.2.1.Mật mã và thông tin 8](#_Toc138104874)

[1.2.2.Hệ mật mã 10](#_Toc138104875)

[1.3. Đề tài nghiên cứu 13](#_Toc138104876)

[1.3.1.Đối tượng nghiên cứu 13](#_Toc138104877)

[1.3.2.Phạm vi nghiên cứu 14](#_Toc138104878)

[1.3.3.Mục tiêu nghiên cứu 14](#_Toc138104879)

[1.3.4.Kiến thực cần có 14](#_Toc138104880)

[1.3.5.Phương pháp nghiên cứu 15](#_Toc138104881)

[Chương 2. Kết quả nghiên cứu 16](#_Toc138104882)

[2.1.Giới thiệu 16](#_Toc138104883)

[2.1.1.Nội dung 16](#_Toc138104884)

[2.1.2.Các yêu cầu cần giải quyết 16](#_Toc138104885)

[2.2.Nội dung thuật toán 16](#_Toc138104886)

[2.2.1.Thuật toán AES 16](#_Toc138104887)

[2.2.1.1. Thuật toán 16](#_Toc138104888)

[2.2.1.2.Xây dựng thuật toán 19](#_Toc138104889)

[2.2.2. Thuật toán Euclid 26](#_Toc138104890)

[2.2.3. Định lý Fermat 33](#_Toc138104891)

[2.2.4. Hàm số Euler 34](#_Toc138104892)

[2.2.5. Thuật toán Miller - Rabin: 36](#_Toc138104893)

[\* Kiểm tra Miller-Rabin 36](#_Toc138104894)

[\* Tiêu chuẩn Miler-Rabin: 37](#_Toc138104895)

[2.2.6. Định lý phần dư Trung Hoa 48](#_Toc138104896)

[2.2.7. Hệ mã hóa công khai RSA. 50](#_Toc138104897)

[2.3.Thiết kế, cài đặt chương trình đề mô thuật toán 55](#_Toc138104898)

[2.3.1.Giao diện chương trình đề mô 55](#_Toc138104899)

[2.3.2.Cài đặt và triển khai 57](#_Toc138104900)

[2.3.2.1.Giới thiệu công cụ 57](#_Toc138104901)

[2.3.2.2. Hướng dẫn cài đặt và chạy chương trình demo đã cài đặt 62](#_Toc138104902)

[2.3.3.Thực hiện bài toán 64](#_Toc138104903)

[2.3.3.1.Phân công công việc 64](#_Toc138104904)

**MỤC LỤC HÌNH ẢNH**

[Hình 1 Các mức độ bảo vệ thông tin 8](#_heading=h.lnxbz9)

[Hình 2 Thông tin gửi đi bị nghe lén 9](#_heading=h.44sinio)

[Hình 3 Bảo vệ thông tin bằng mật mã 9](#_heading=h.2jxsxqh)

[Hình 4 Quá trình mã hóa và giải mã thông tin 10](#_heading=h.4i7ojhp)

[Hình 5 Sơ đồ mã hóa và giải mã bằng khóa riêng 12](#_heading=h.2xcytpi)

[Hình 6 Sơ đồ mã hóa và giải mã bằng khóa công khai 12](#_heading=h.1ci93xb)

[Hình 7 Sơ đồ mã hóa AES 128 bits 18](#_heading=h.1hmsyys)

[Hình 8 bảng giá trị S-Box 19](#_heading=h.2grqrue)

[Hình 9 Bảng giá trị S-box nghịch đảo 19](#_heading=h.vx1227)

[Hình 10 Quá trình mã hóa 20](#_heading=h.3fwokq0)

[Hình 11 Các dạng đồ thi đường cong 25](#_heading=h.2u6wntf)

[Hình 12 Sơ đồ truyền tải đề thi qua mạng internet 34](#_heading=h.1mrcu09)

[Hình 13 Demo chương trình bằng ngôn ngữ Pyth của Mai Thị Khánh Linh 35](#_heading=h.1jlao46)

[Hình 14 Demo chương trình bằng ngôn ngữ C# của Trần Đăng Khoa 35](#_heading=h.2zbgiuw)

[Hình 15Demo chương trình bằng ngôn ngữ Python của Bùi Trung Kiên 36](#_heading=h.1egqt2p)

[Hình 16 Demo chương trình bằng ngôn ngữ Java của Nguyễn Viết Khánh 36](#_heading=h.3ygebqi)

[Hình 17 Ngôn ngữ lập trình PHP 37](#_heading=h.2dlolyb)

[Hình 18 Ngôn ngữ lập trình C# 38](#_heading=h.2dlolyb)

[Hình 19 Ngôn ngữ lập trình Java 39](#_heading=h.2dlolyb)

[Hình 20 Ngôn ngữ lập trình Python 40](#_heading=h.2dlolyb)

Hình 21 Ngôn ngữ lập trình Javascript

**LỜI CẢM ƠN**

*Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy* ***TS. Nguyễn Bá Nghiễn****. Trong quá trình tìm hiểu và học tập bộ môn An toàn và bảo mật thông tin, chúng em đã được thầy cung cấp và truyền đạt tất cả kiến thức chuyên môn cần thiết và quý giá nhất. Ngoài ra, chúng em còn được rèn luyện một tinh thần học tập và làm việc độc lập , sáng tạo. Đây là tính cách hết sức cần thiết để có thể thành công khi bắt tay vào nghề nghiệp trong tương lai. Nhóm đã nhận được sự hướng dẫn và những chia sẻ rất tận tình, tâm huyết của thầy. Từ những hướng dẫn tận tình của thầy cùng với kiến thức mà nhóm đã học tập, tìm hiểu, chúng em đã hoàn thành báo cáo đề tài “****Ứng dụng mã hóa lai vào công tác bảo mật trong truyền tải đề thi****”.Đây làcơ hội để nhóm em có thể áp dụng, tìm hiểu thêm và tổng kết lại những kiến thức mà mình đã học. Đồng thời, rút ra được những kinh nghiệm thực tế và quý giá trong suốt quá trình thực hiện đề tài.*

*Với tất cả sự cố gắng, nỗ lực của mình, nhóm em đã hoàn thành tốt nhất bài báo cáo này.Trong quá trình thực hiện đề tài, do kiến thức còn nhiều hạn chế và thiếu sót, nhóm chúng emchắc chắn không tránh khỏinhững thiếu sót khi thực hiện . Nhóm chúng em mong nhận được sự góp ý của thầy để đề tài cũng như ứng dụng demo của các thành viên trong nhóm được đầy đủ và hoàn thiện hơn về kiến thức cũng như các chức năng của ứng dụng đề mô.Sự phê bình, góp ý của thầy sẽ là những bài học kinh nghiệm rất quý báu cho công việc thực tế của chúng em sau này .*

*Kính chúc thầy thật nhiều sức khoẻ, hạnh phúc và thành công trong cuộc sống.*

*Nhóm chúng em xin chân thành cảm ơn!*

**LỜI NÓI ĐẦU**

Với sự bùng nổ mạnh của công nghệ thông tin và sự phát triển của mạng Internet nên việc trao đổi thông tin trở nên dễ dàng hơn bao giờ hết. Tuy nhiên, phát sinh thêm một vấn đề ngày càng trở nên cấp bách và cần thiết về yêu cầu an toàn mạng, an ninh dữ liệu, bảo mật thông tin trong môi trường mạng cũng như trong thực tiễn.

Trên thế giới có nhiều quốc gia và nhà khoa học nghiên cứu vẫn đề bảo mật, đưa ra nhiều thuật toán giúp thông tin không bị đánh cắp hoặc nếu bị lấy cắp cũng không sử dụng được. Trong các giải pháp đó là an toàn thông tin bằng mật mã. Ở đề tài này nhóm em đề cập tới thuật toán mã hóa AES (Advanced Encryption Standard) từng được chính phủ Mỹ và nhiều quốc gia trên thế giới sử dụng. Đến giờ, AES vẫn được dùng cho các tài liệu tuyệt mật, được cho là FIPS (Federal Information Processing Standard - tiêu chuẩn xử lý thông tin liên bang). Sau đó nó được dùng trong khối tư nhân, là chuẩn mã hóa phổ biến nhất với mã hóa khóa đối xứng.

Bên cạnh đó mật mã RSA là kĩ thuật mã khóa công khai, giúp tạo mật mã nhanh hơn, nhỏ hơn và mạnh hơn. Để ứng dụng 2 phương pháp trên vào thực tiễn, được sự hướng dẫn của thầy Nguyễn Bá Nghiễn, chúng em lựa chọn đề tài ***“Ứng dụng hệ thống mã hóa lai vào công tác bảo mật trong truyền tải đề thi”***với mong muốn áp dụng kiến thức đã học, giải quyết bài toán bảo mật .

Bảo mật đề thi có vai trò hết sức quan trọng đối với các kỳ thi.Đề thi là một trong những tài liệu mật của quốc gia. Hằng năm, các trường học phải thường xuyên tổ chức các kỳ thi nhằm tuyển chọn học sinh vào trường, kỳ thi đánh giá kết quả học tập của học sinh như: Thi tuyển sinh đầu vào, kiểm tra chất lượng, thi học kỳ, thi tốt nghiệp, thi học sinh giỏi… Trong các kỳ thi đó, có những đợt thi các trường thi chung đề thi của Bộ Giáo dục và Đào tạo, của Sở Giáo dục và Đào tạo (SGD&ĐT). Hiện nay, SGD&ĐT bảo mật đề thi của các kỳ thi bằng cách niêm phong các túi đề thi.

Việc bố trí nhân sự, in sao đề thi sẽ thực hiện theo quy định. Phương án vận chuyển bàn giao đề thi từ địa điểm in sao đến các điểm thi được tính đến, bao gồm cả kế hoạch dự phòng. Ban vận chuyển và bàn giao đề thi nhận các túi đề thi còn nguyên niêm phong từ Ban in sao đề thi bảo quản, vận chuyển, phân phối đề thi đến các điểm thi. Các túi đề thi phải được bảo quản trong hòm sắt được khóa, niêm phong và bảo vệ 24 giờ/ngày. Tại các điểm thi, đề thi và bài thi được để trong các tủ riêng biệt. Tủ đựng đề thi, bài thi đảm bảo chắc chắn, được khóa và niêm phong (nhãn niêm phong có đủ chữ ký của trưởng điểm thi, thanh tra và công an), chìa khóa do trưởng điểm thi giữ. Khi mở niêm phong phải có chứng kiến của những người ký nhãn niêm phong, lập biên bản ghi rõ thời gian mở, lý do mở, tình trạng niêm phong.

Ngoài ra, khu vực bảo quản đề thi sẽ có công an trực, bảo vệ liên tục 24 giờ/ngày và phải bảo đảm an toàn phòng chống cháy, nổ. Phòng bảo quản đề thi bảo đảm an toàn, chắc chắn; có camera an ninh giám sát, ghi hình các hoạt động tại phòng liên tục; công an trực, bảo vệ liên tục 24 giờ/ngày; có một phó trưởng điểm thi là người của trường phổ thông không có thí sinh dự thi tại điểm thi trực tại phòng trong suốt thời gian đề thi, bài thi được lưu tại điểm thi. Từng hội đồng thi có trách nhiệm lập phương án bảo vệ đề thi trong suốt quá trình tổ chức kỳ thi. Với việc nhận và chuyển đề thi theo phương thức này có thể gặp nhiều trở ngại cũng như việc đảm bảo an toàn, bí mật cho đề thi chứa đựng nhiều yếu tố rủi ro, kinh phí cho việc giao nhận và bảo vệ đề thi rất tốn kém.

Để góp phần khắc phục một phần những hạn chế trên, việc sử dụng các công cụ của mật mã học ứng dụng vào công tác bảo mật đề thi trong truyền tải đề thi qua mạng là một vấn đề mang tính thời sự và cấp thiết.

# Chương 1. Tổng quan

## 1.1. Tổng quan về an toàn và bảo mật thông tin

### 1.1.1.Sự cần thiết của việc đảm bảo an toàn thông tin

Ngày nay sự xuất hiện của internet toàn cầu đã giúp cho việc trao đổi thông tin trở nên nhanh gọn, dễ dàng. Các phương thức chia sẻ dữ liệu qua mạng làm cho việc trao đổi, mua bán, chuyển tiền, … diễn ra mỗi ngày trên nền tảng số.

Tuy nhiên vấn đề mới lại phát sinh. Những thông tin đang nằm ở kho dữ liệu hay đang được truyền đi có thể bị trộm cắp, bị làm sai lệch, giả mạo. Điều này làm ảnh hưởng đến độ an toàn của thông tin nhạy cảm, tin mật, … có thể tác động lớn đến nhiều cá nhân, tổ chức, hay ác động đến an ninh quốc gia.

### 1.1.2.Khái niệm an toàn thông tin

Định nghĩa của an toàn thông tin được nêu ra từ nhiều nguồn khác nhau, chúng ta có thể hiểu theo nhiều cách sau: *An toàn thông tin nghĩa là thông tin được bảo vệ, các hệ thống và dịch vụ có khả năng chống lại những sự can thiệp, lỗi và những tai họa không mong đợi, các thay đổi tác động đến độ an toàn của hệ thống là nhỏ nhất. Hệ thống không an toàn là hệ thống tồn tại những điểm: thông tin bị rò rỉ ra ngoài, thông tin bị thay đổi,* ...

Giá trị thực sự của thông tin chỉ đạt được khi thông tin được cung cấp chính xác và kịp thời, hệ thống phải hoạt động chuẩn xác thì mới có thể đưa ra những thông tin có giá trị cao. *Mục tiêu của an toàn bảo mật trong công nghệ thông tin là đưa ra một số tiêu chuẩn an toàn và áp dụng các tiêu chuẩn an toàn này vào chỗ thích hợp để giảm bớt và loại trừ những nguy hiểm có thể xảy ra.* Ngày nay với kỹ thuật truyền nhận và xử lý thông tin ngày càng phát triển và phức tạp nên hệ thống chỉ có thể đạt tới một mức độ an toàn nào đó và không có một hệ thống an toàn tuyệt đối.

### 1.1.3.Các yêu cầu an toàn bảo mật thông tin

Khi nhu cầu trao đổi thông tin dữ liệu ngày càng lớn và đa dạng, các tiến bộ về điện tử - viễn thông và công nghệ thông tin không ngừng được phát triển ứng dụng để nâng cao chất lượng và lưu lượng truyền tin thì các quan niệm ý tưởng và biện pháp bảo vệ thông tin dữ liệu cũng được đổi mới. Bảo vệ an toàn thông tin dữ liệu là một chủ đề rộng, có liên quan đến nhiều lĩnh vực và trong thực tế có thể có rất nhiều phương pháp được thực hiện để bảo vệ an toàn thông tin dữ liệu. Các phương pháp bảo vệ an toàn thông tin dữ liệu có thể được quy tụ vào ba nhóm sau:

- Bảo vệ an toàn thông tin bằng các biện pháp hành chính.

- Bảo vệ an toàn thông tin bằng các biện pháp kỹ thuật (phần cứng).

- Bảo vệ an toàn thông tin bằng các biện pháp thuật toán (phần mềm).

Ba nhóm trên có thể được ứng dụng riêng rẽ hoặc phối kết hợp. Môi trường khó bảo vệ an toàn thông tin nhất và cũng là môi trường đối phương dễ xâm nhập nhất đó là môi trường mạng và truyền tin. Biện pháp hiệu quả nhất và kinh tế nhất hiện nay trên mạng truyền tin và mạng máy tính là biện pháp thuật toán.  
Biện pháp hiệu quả nhất và kinh tế nhất hiện nay trên mạng truyền tin và mạng máy tính là biện pháp thuật toán.

Ngày nay, với sự phát triển rất nhanh của khoa học công nghệ, các biện pháp tấn công ngày càng tinh xảo hơn, độ an toàn của thông tin có thể bị đe dọa từ nhiều nơi, theo nhiều cách khác nhau, chúng ta cần phải đưa ra các chính sách đề phòng thích hợp. Các yêu cầu cần thiết của việc bảo vệ thông tin và tài nguyên:

* + *Đảm bảo bí mật (Bảo mật):* thông tin không bị lộ đối với người không được phép.
  + *Đảm bảo tính tin cậy (Confidentiality):* Thông tin và tài nguyên không thể bị truy cập trái phép bởi những người không có quyền hạn.
  + *Đảm bảo tính toàn vẹn (Integrity):* Thông tin và tài nguyên không thể bị sửa đổi, bị thay thế bởi những người không có quyền hạn.
  + *Đảm bảo tính sẵn sàng (Availability):* Thông tin và tài nguyên luôn sẵn sàng để đáp ứng sử dụng cho người có quyền hạn.
  + *Đảm bảo tính không thể chối bỏ (Non-repudiation):* Thông tin và tài nguyên được xác nhận về mặt pháp luật của người cung cấp.

**Các nội dung an toàn thông tin**

* *Nội dung chính:*
  + An toàn máy tính: là sự bảo vệ các thông tin cố định bên trong máy tính, là khoa học về bảo đảm an toàn thông tin trong máy tính
  + An toàn truyền tin: là sự bảo vệ thông tin trên đường truyền tin(thông tin được truyền từ hệ thống này sang hệ thống khác), là khoa học bảo đảm an toàn thông tin trên đường truyền tin.
* *Nội dung chuyên ngành:*
  + An toàn dữ liệu (data security)
  + An toàn cơ sở dữ liệu (database security)
  + An toàn hệ điều hành (operation system security)
  + An toàn mạng máy tính (network security)

**Các chiến lược bảo đảm an toàn thông tin**

*Giới hạn quyền hạn tối thiểu (Last Privilege):*theo nguyên tắc này bất kỳ một đối tượng nào cùng chỉ có những quyền hạn nhất định đối với tài nguyên mạng.

*Bảo vệ theo chiều sâu (Defence In Depth):* Không nên dựa vào một chế độ an toàn nào dù cho chúng rất mạnh, mà nên tạo nhiều cơ chế an toàn để tương hỗ lẫn nhau.

*Nút thắt (Choke Point):* Tạo ra một “cửa khẩu” hẹp, và chỉ cho phép thông tin đi vào hệ thống của mình bằng con đường duy nhất chính là “cửa khẩu” này.

*Điểm nối yếu nhất (Weakest Link):*Chiến lược này dựa trên nguyên tắc: “ Một dây xích chỉ chắc tại mắt duy nhất, một bức tường chỉ cứng tại điểm yếu nhất”.

*Tính toàn cục:* Các hệ thống an toàn đòi hỏi phải có tính toàn cục của các hệ thống cục bộ.

*Tính đa dạng bảo vệ:* Cần phải sử dụng nhiều biện pháp bảo vệ khác nhau cho hệ thống khác nhau, nếu không có kẻ tấn công vào được một hệ thống thì chúng cũng dễ dàng tấn công vào các hệ thống khác.

### 1.1.4.Các phương pháp bảo vệ thông tin

*Quyền truy nhập:* Là lớp bảo vệ trong cùng nhằm kiểm soát các tài nguyên của mạng và quyền hạn trên tài nguyên đó.

*Đăng ký tên /mật khẩu:* Thực ra đây cũng là kiểm soát quyền truy nhập, nhưng không phải truy nhập ở mức thông tin mà ở mức hệ thống.

*Mã hoá dữ liệu:* Dữ liệu bị biến đổi từ dạng nhận thức được sang dạng không nhận thức được theo một thuật toán nào đó và sẽ được biến đổi ngược lại ở trạm nhận (giải mã).

*Bảo vệ vật lý:* Ngăn cản các truy nhập vật lý vào hệ thống.

*Tường lửa:* Ngăn chặn thâm nhập trái phép và lọc bỏ các gói tin không muốn gửi hoặc nhận vì các lý do nào đó để bảo vệ một máy tính hoặc cả mạng nội bộ (intranet).

*Quản trị mạng:* Công tác quản trị mạng máy tính phải được thực hiện một cách khoa học. Toàn bộ hệ thống hoạt động bình thường trong giờ làm việc. Backup dữ liệu quan trọng theo định kỳ, có kế hoạch bảo trì định kỳ, bảo mật dữ liệu, phân quyền, …

Diagram, table

Description automatically generated

*Hình 1 Các mức độ bảo vệ thông tin*

## 1.2. An toàn thông tin bằng mật mã

### 1.2.1.Mật mã và thông tin

Mật mã là một ngành khoa học chuyên nghiên cứu các phương pháp truyền tin bí mật. Mật mã bao gồm : Lập mã và phá mã.

* *Lập mã* bao gồm hai quá trình: mã hóa và giải mã. Các sản phẩm của lĩnh vực này là các hệ mã mật , các hàm băm, các hệ chữ ký điện tử, các cơ chế phân phối, quản lý khóa và các giao thức mật mã.
* *Phá mã:* Nghiên cứu các phương pháp phá mã hoặc tạo mã giả. Sản phẩm của lĩnh vực này là các phương pháp phá mã , các phương pháp giả mạo chữ ký, các phương pháp tấn công các hàm băm và các giao thức mật mã.

Diagram, schematic

Description automatically generated

*Hình 2 Thông tin gửi đi bị nghe lén*

Để bảo vệ thông tin trên đường truyền người ta thường biến đổi nó từ dạng nhận thức được sang dạng không nhận thức được trước khi truyền đi trên mạng, quá trình này được gọi là mã hoá thông tin (encryption), ở trạm nhận phải thực hiện quá trình ngược lại, tức là biến đổi thông tin từ dạng không nhận thức được (dữ liệu đã được mã hoá) về dạng nhận thức được (dạng gốc), quá trình này được gọi là giải mã (decryption). Đây là một lớp bảo vệ thông tin rất quan trọng và được sử dụng rộng rãi trong môi trường mạng.

Diagram

Description automatically generated

*Hình 3 Bảo vệ thông tin bằng mật mã*

Để bảo vệ thông tin bằng mật mã người ta thường tiếp cận theo hai hướng:

* *Theo đường truyền (Link\_Oriented\_Security):* thông tin được mã hoá để bảo vệ trên đường truyền giữa hai nút mà không quan tâm đến nguồn và đích của thông tin đó. Thông tin chỉ được bảo vệ trên đường truyền, tức là ở mỗi nút đều có quá trình giải mã sau đó mã hoá để truyền đi tiếp, do đó các nút cần phải được bảo vệ tốt.
* *Từ nút đến nút (End\_to\_End):* thông tin trên mạng được bảo vệ trên toàn đường truyền từ nguồn đến đích. Thông tin sẽ được mã hoá ngay sau khi mới tạo ra và chỉ được giải mã khi về đến đích. Cách này mắc phải nhược điểm là chỉ có dữ liệu của người dùng thì mới có thể mã hóa được còn dữ liệu điều khiển thì giữ nguyên để có thể xử lý tại các nút.

### 1.2.2.Hệ mật mã

**Vai trò của hệ mật mã**

Các hệ mật mã phải thực hiện được các vai trò sau:

* Hệ mật mã phải che dấu được nội dung của văn bản rõ (PlainText) để đảm bảo sao cho chỉ người chủ hợp pháp của thông tin mới có quyền truy cập thông tin (Secrety), hay nói cách khác là chống truy nhập không đúng quyền hạn.
* Tạo các yếu tố xác thực thông tin, đảm bảo thông tin lưu hành trong hệ thống đến người nhận hợp pháp là xác thực (Authenticity).
* Tổ chức các sơ đồ chữ ký điện tử, đảm bảo không có hiện tượng giả mạo, mạo danh để gửi thông tin trên mạng.

Ưu điểm lớn nhất của bất kỳ hệ mật mã nào đó là có thể đánh giá được độ phức tạp tính toán mà “kẻ địch” phải giải quyết bài toán để có thể lấy được thông tin của dữ liệu đã được mã hoá. Tuy nhiên mỗi hệ mật mã có một số ưu và nhược điểm khác nhau, nhưng nhờ đánh giá được độ phức tạp tính toán mà ta có thể áp dụng các thuật toán mã hoá khác nhau cho từng ứng dụng cụ thể tuỳ theo yêu cầu về độ an toàn.

**Mã hóa và giải mã thông tin**

Graphical user interface

Description automatically generated

*Hình 4 Quá trình mã hóa và giải mã thông tin*

* Mã hóa: Quá trình chuyển đổi dữ liệu gốc thành dữ liệu được mã hóa sao cho người khác không thể đọc hiểu được.
* Giải mã: Là quá trình ngược lại của mã hóa, biến đổi dữ liệu đã được mã hóa thành dạng gốc ban đầu.
* Bản mã: Tệp dữ liệu đã được mã hóa.

Một hệ thống mã hóa bao gồm các thành phần sau:

* PlainText : Bản tin sẽ được mã hóa hay bản tin gốc.
* CipherText : Bản tin đã được mã hóa hay bản tin mã.
* Thuật toán mã hóa và giải mã :
  + - Encryption : quá trình chuyển bản tin gốc sang dạng mật mã.
    - Decryption : quá trình giải bản tin dạng mật mã trở về bản tin gốc.
    - Cách chọn khóa : giá trị toán học dùng để thực hiện mã hóa.

Nhiều phương pháp mã hóa đã được đưa ra dựa trên những giải thuật toán phức tạp, để tạo khó khăn cho những ai đó muốn phá mật mã mà không cần được ai trao chìa khóa. Nói tạo khó khăn là vì trên lý thuyết ta không thể nói việc tìm chìa khóa là vô phương. Nhưng nếu trở ngại đủ lớn để làm nản lòng kẻ gian thì đã là một mức độ an toàn tốt. Quá trình mã hóa và giải mã có thể được minh họa theo sơ đồ sau:

**Các thành phần của một hệ mật mã**

Một hệ mã mật là bộ 5 (P, C, K, E, D) thoả mãn các điều kiện sau:

* P là tập hữu hạn các bản rõ (PlainText), nó được gọi là không gian bản rõ chứa bản tin gốc ban đầu.
* C là tập hợp hữu hạn bản mã (Crypto), nó còn được gọi là không gian các bản mã. Một phần tử của C có thể nhận được bằng cách áp dụng phép mã hóa EK lên một phần tử P, với k ∈ K.
* K là tập hữu hạn các khóa hay còn gọi là không gian khoá. Đối với mỗi phần tử k của K được gọi là một khoá (Key). Số lượng của không gian khoá phải đủ lớn để “kẻ địch” không có đủ thời gian thử mọi khoá có thể (phương pháp vét cạn).
* Đối với mỗi k ∈ K có một quy tắc mã  eK: P → C  và một quy tắc giải mã tương ứng dK∈ D. Mỗi eK: P→ C và dK: C → P là những hàm mà:

dK (eK(x))=x với mọi bản rõ x ∈ P.

* Hàm giải mã dk chính là ánh xạ ngược của hàm mã hóa ek

**Phân loại hệ mật mã**

Có nhiều cách để phân loại hệ mật mã. Dựa vào cách truyền khóa có thể phân các hệ mật mã thành hai loại:

* Hệ mật đối xứng (hay còn gọi là mật mã khóa bí mật)
* Hệ mật mã bất đối xứng (hay còn gọi là mật mã khóa công khai)

Ngoài ra nếu dựa vào thời gian đưa ra hệ mật mã ta còn có thể phân làm hai loại: Mật mã cổ điển (là hệ mật mã ra đời trước năm 1970) và mật mã hiện đại (ra đời sau năm 1970). Còn nếu dựa vào cách thức tiến hành mã thì hệ mật mã còn được chia làm hai loại là mã dòng (tiến hành mã từng khối dữ liệu, mỗi khối lại dựa vào các khóa khác nhau, các khóa này được sinh ra từ hàm sinh khóa, được gọi là dòng khóa ) và mã khối (tiến hành mã từng khối dữ liệu với khóa như nhau).

**Mã hóa bằng khóa bí mật**

Các hệ thống mã hóa với khóa bí mật còn được gọi là mã hóa bằng khóa riêng, mã hóa đối xứng sử dụng duy nhất một khóa cho cả quá trình mã hóa lẫn quá trình giải mã.

* Stream Algorithms/Stream Ciphers: các thuật toán hoạt động trên văn bản bình thường theo từng bit một.
* Block Algorithms/Block Ciphers : các thuật toán hoạt động trên văn bản theo các khối (32 bit, 64 bit, 128 bit, ...).
* Một số thuật toán đang được sử dụng rộng rãi hiện nay : DES, TripleDES, RC5, RC6, Rijndael ...

Quá trình mã hóa và giải mã bằng cách sử dụng khóa bí mật được minh họa như hình dưới đây:

*Diagram, schematic

Description automatically generated*

*Hình 5 Sơ đồ mã hóa và giải mã bằng khóa riêng*

**Mã hóa bằng khóa công khai**

Diagram, schematic

Description automatically generated

*Hình 6 Sơ đồ mã hóa và giải mã bằng khóa công khai*

Mã hóa bằng khóa công khai còn gọi là mã hóa bất đối xứng hay mã hóa bằng khóa chung. Sự khác biệt cơ bản giữa một hệ thống mã hóa bằng khóa bí mật với hệ thống mã hóa bằng khóa công khai là hệ thống mã hóa khóa công khai dùng hai khóa khác nhau để mã hóa và giải mã. Do đó, một bộ mã công khai sẽ bao gồm hai khóa: một khóa dành cho người mã hóa thường được công khai, và khóa còn lại dùng cho người giải mã thường được giữ bí mật. Như vậy, hệ thống mã hóa với khóa công khai cần có một quá trình sinh ra hai khóa để mã hóa và giải mã thông điệp. Các khóa này được xem như là một đôi:

* Public-key (khóa công khai): được phép công khai mà không phải chịu rủi ro về an toàn. Khóa này được dùng để mã hóa thông điệp.
* Private-key (khóa bí mật): không được để lộ. Mỗi thông điệp được mã hóa bằng public-key chỉ có thể giải mã bằng một khóa mật thích hợp.
* Một số thuật toán mã hóa công khai phổ biến : RSA, Diffie-Hellman KeyExchange Algorithm (dùng cho việc phân phối và trao đổi khóa).

Như vậy, với sự bùng nổ của mạng toàn cầu mọi hệ thống thông tin đều phải đương đầu với bài toán an toàn và bảo mật. Như đã trình bày, có nhiều chiến lược cũng như phương pháp bảo đảm bảo an toàn thông tin. Trong đó, an toàn thông tin bằng mật mã có vai trò pháp quan trọng và được ứng dụng rộng khắp không chỉ trong ngành công nghệ thông tin mà còn dùng để bảo mật những thông tin và tài liệu quan trọng ngoài đời. (Ví dụ như bảo mật đề thi trong tuyển sinh được đề cập đến trong chương sau)

## 1.3. Đề tài nghiên cứu

Đề tài: *Ứng dụng mã hóa lai vào công tác bảo mật trong truyền tải đề thi*

### 1.3.1.Đối tượng nghiên cứu

Nghiên cứu về chương trình mã hóa và giải mã AES trong mật mã học, hệ mã hóa RSA.

An toàn thông tin là bảo vệ các đặc tính riêng tư (confidentialy), toàn vẹn (intergrity) và khả dụng (availabity) của thông tin.

* C: (Confidentialy) bảo vệ tính riêng tư của dữ liệu thông qua các cơ chế chứng thực và mã hóa, ngăn ngừa những người không hợp lệ sẽ không được đọc những thông tin. Giống như các bì thư khi phát lương thưởng được dán chữ Confidentialy, chúng ta có thể hình dung trong môi trường công nghệ thông tin là một người chưa đăng nhập vào Domain sẽ không được truy cập những dữ liệu chỉ chia sẻ cho các Domain User.
* I: (Intergrity) bảo vệ tính toàn vẹn của dữ liệu thông qua các thuật toán RSA, SHA, MD5 ... ngăn ngừa attacker thay đổi các thông tin nhạy cảm trong quá trình truyền.
* A: (Available) bảo đảm dữ liệu luôn ở trong trạng thái sẵn sằng đáp ứng nhu cầu của người dùng.
* Non-Repudiation: Tính không thể chối bỏ, nghĩa là dữ liệu người nào gửi đi thì họ phải có trách nhiệm với các thông tin của mình thông qua các xác nhận nguồn gốc như chữ kí điện tử.

Các bài báo đầu tiên của Diffie và Hellman đã đưa ra một cách tiếp cận mới đối với mật mã, và thực tế đã thách thức các nhà mật mã học để đưa ra một thuật toán mật mã đáp ứng các yêu cầu đối với các hệ thống khóa công khai. Một số thuật toán đã được đề xuất cho mật mã khóa công khai. Một số trong số này ban đầu đầy hứa hẹn sau đó nó lại có thể bị bẻ gãy.

### 1.3.2.Phạm vi nghiên cứu

Các kì thi lớn cần đảm bảo mức an toàn thông tin cao như các cuộc thi THPTQG hay cuộc thi cấp tỉnh và miền.

Có thể ứng dụng sang một số linh vực khác như truyền tải tin tình báo, thư mật và các tài liệu quan trong,...

### 1.3.3.Mục tiêu nghiên cứu

* Xác định và làm rõ các vấn đề liên quan đến đề tài.
* Phân tích chương trình mã hoá và giải mã AES.
* Phân tích hệ mã hóa RSA.
* Ứng dụng cài đặt chương trình mã hóa đề thi với các ngôn ngữ lập trình khác nhau.
* Xác định các ứng dụng thực tế sử dụng chương trình.

### 1.3.4.Kiến thực cần có

**Hệ mật mã đối xứng** còn gọi là mật mã khoá đơn hoặc là mật mã khoá riêng. Trong các hệ mật mã này, khoá mật mã mã hoá bảo mật giống với khoá giải mã hoặc trên thực tế là cùng đẳng cấp. Lúc này khoá mật mã cần phải có một đường truyền an toàn để truyền đưa khoá mật mã từ phía người truyền cho phía người nhận. Đặc điểm của mật mã đối xứng là bất luận khi gia công bảo mật hay là khi giải mã đều sử dụng cùng một khoá mật mã. Do đó tính an toàn của mật mã này là sự an toàn của khoá mật mã. nếu như khoá mật mã bị tiết lộ, thì hệ thống mật mã này sẽ bị phá vỡ.

**Hệ mật mã bất đối xứng** còn gọi là mật mã khoá công khai hoặc mật mã khoá đôi. Trong các hệ mật mã này quá trình mã hoá và giải mã có chìa khoá khác nhau, lúc này không cần có đường truyền an toàn để truyền đưa khoá mật mã mà chỉ cần bộ phát sinh khoá mã tại chỗ để tạo ra khoá giải mã đồng thời lấy đó để khống chế các thao tác giải mã.

Thuật toán **mật mã khóa công khai** được dựa trên các chức năng toán học thay vì thay thế và hoán vị. Quan trọng hơn, mật mã khóa công khai là không đối xứng bao gồm việc sử dụng hai khóa riêng biệt, trái ngược với mã hóa đối xứng, chỉ sử dụng một khóa. Trước khi tiến hành, ta nên đề cập đến một số khái niệm sai lầm phổ biến liên quan đến mã hóa khóa công khai. Một quan niệm sai lầm như vậy là: Mật mã khóa công khai an toàn hơn từ việc phân tích mật mã hơn mã hóa đối xứng. Trong thực thế, bảo mật của bất kì chương trình mã hóa nào phụ thuộc vào độ dài của khóa và công việc tính toán liên quan đến phã vỡ một mật mã. Không có gì về nguyên tắc của mã hóa đối xứng hoặc mã hóa khóa công khai mà làm cho những cấp trên khác chống lại phân tích mật mã.

### 1.3.5.Phương pháp nghiên cứu

* Tìm kiếm tài liệu trên internet.
* Quản lý và lưu trữ tài liệu nghiên cứu trên cloud.
* Sử dụng các IDE Eclipse, Netbeans, Visual Studio, VSCode để xây dựng chương trình.
* Sử dụng các ngôn ngữ lập trình Dart, Java, C#, Python, Javascript.

# Chương 2. Kết quả nghiên cứu

Như đã trình bày ở chương trước: mọi thông tin cần bảo vệ nên được mã hóa. Với sự phát triển của công nghệ, nhiều hệ mật mã khác nhau được ra đời từ các hệ mật mã truyền thống đến các hệ mật mã hiện đại. Từ những năm 70 của thế kỷ trước, các nhà khoa học đã nghiên cứa và tạo ra nhiều phương thức mật mã với tốc độ mã hóa rất nhanh chỉ cần giữ bí mật khóa mã (mã hóa đối xứng) và mã hóa được mọi dữ liệu tùy ý. Đó là một bước tiến vĩ đại của kỹ thuật mật mã .Trong đó mã AES (Advanced Encryption Standard) và RSA là 2 chuẩn mã hóa rất phổ biến.

## 2.1.Giới thiệu

- Khi hoàn thiện xong chương trình thì sẽ thực hiện được các yêu cầu về chương  trình mà đề tài đề ra như mã hóa AES, mã hóa RSA.

- Đề tài : Ứng dụng hệ thống mã hóa lai vào công tác bảo mật trong truyền tải đề thi.

### 2.1.1.Nội dung

1. Tìm hiểu mật mã AES.
2. Tìm hiểu mật mã RSA
3. Ứng dụng xây dựng chương trình demo.
4. Demo chương trình

### 2.1.2.Các yêu cầu cần giải quyết

1. Đọc tài liệu và hiểu được vấn đề đặt ra,nắm được các kiến thức về mã hóa AES và RSA một cách thành thạo(cả tiếng việt và tiếng anh).
2. Hiểu được vấn đề của bài toán.
3. Đọc hiểu được một số tài liệu chuyên môn bằng tiếng Anh
4. Nắm vững một ngôn ngữ lập trình cơ bản(Java,C#,C++,Python)và giải được bài toán có tính ứng dụng vào thực tiễn.

## 2.2.Nội dung thuật toán

### 2.2.1.Thuật toán AES

#### 2.2.1.1. Thuật toán

AES là thuật toán mã hóa khối, nó làm việc với các khối dữ liệu 128bit và độ dài khóa 128bit, 192bit hoặc 256bit. Các khóa mở rộng sử dụng trong chu trình được tạo ra bởi thủ tục sinh khóa Rijndael. Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu đầu vào 128bit được chia thành 16byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay một ma trận 4x4 của các byte, nó gọi là ma trận trạng thái.

Thuật toán AES khá phức tạp, được mô tả khái quát gồm 3 bước như sau:

* 1 Vòng khởi tạo chỉ gồm phép AddRoundKey
* Nr -1 Vòng lặp gồm 4 phép biển đổi lần lượt: SubBytes, ShiftRows, MixColumns, AddRoundKey.
* 1 Vòng cuối gồm các phép biến đổi giống vòng lặp và không có phép MixColumns.

**Khái quát:**

*1. Mở rộng khóa* - Các khóa phụ dùng trong các vòng lặp được sinh ra từ khóa chính AES sử dụng thủ tục sinh khóa Rijndael.

*2. InitialRound - AddRoundKey*— Mỗi byte trong state được kết hợp với khóa phụ sử dụng XOR

*3. Rounds - SubBytes*—bước thay thế phi tuyến tính, trong đó mỗi byte trong state được thay thế bằng một byte khác sử dụng bảng tham chiếu - ShiftRows—bước đổi chỗ, trong đó mỗi dòng trong state được dịch một số bước theo chu kỳ - MixColumns—trộn các cột trong state, kết hợp 4 bytes trong mỗi cột - AddRoundKey

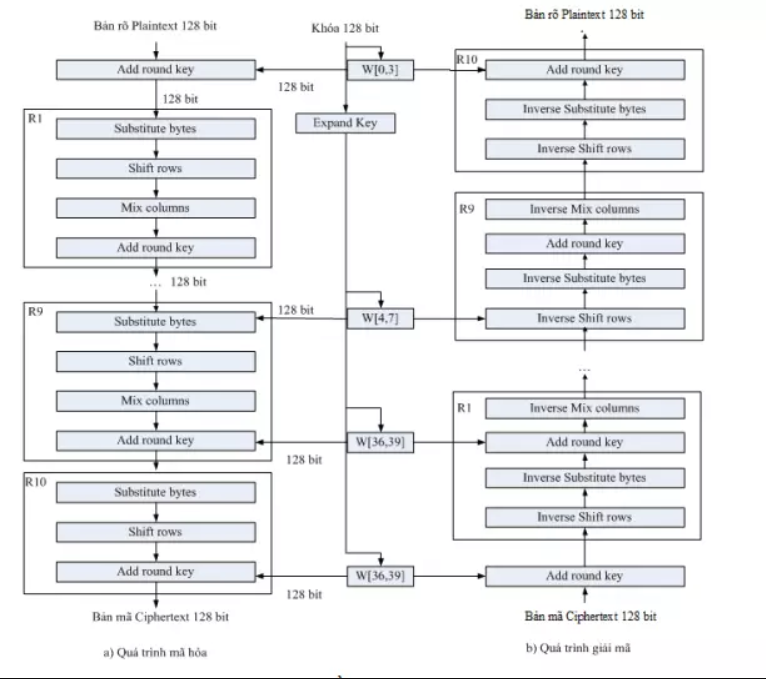
*4. Final Round (không MixColumns) - SubBytes - ShiftRows - AddRoundKey*.

Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm sử dụng là 4 hàm ngược của quá trình mã hóa. Riêng đối với cấu trúc giải mã trong AES gồm 2 chế độ giải mã:

- Ở cấu trúc giải mã ngược, gồm vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi tạo chỉ có phép biến đổi AddRounKey, vòng lặp gồm lần lượt 4 phép biến đổi chính: InvShiftRows, InvSubBytes, AddRounKey, InvMixColumns; vòng kết thúc khác với vòng lặp chính ở chỗ không có phép InvMixColumns.

- Ngược lại với cấu trúc giải mã ngược là cấu trúc giải mã xuôi, việc ngược lại thể hiện ở điểm: trong cấu trúc giải mã xuôi việc sắp xếp các phép biến đổi ngược giống hệt với cấu trúc mã hóa, cụ thể bao gồm: vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc.

Trong đó vòng khởi là phép AddRounKey; ở vòng lặp thứ tự các phép biến đổi ngược lần lượt là: InvSubBytes, InvShiftRows, InvMixColumns, AddRounKey; vòng kết thúc giống vòng lặp nhưng được lược bỏ phép InvMixColumns. Một điểm khác biệt nữa trong hai cấu trúc giải mã ngược và giải mã xuôi đó là: Trong giải mã ngược khóa vòng giải mã chính là khóa vòng mã hóa với thứ tự đảo ngược. Còn trong giải mã xuôi thì khóa giải mã ngoài việc đảo ngược thứ tự khóa vòng mã hóa còn phải thực hiện phép InvMixColumns đối với các khóa vòng của vòng lặp giải mã.



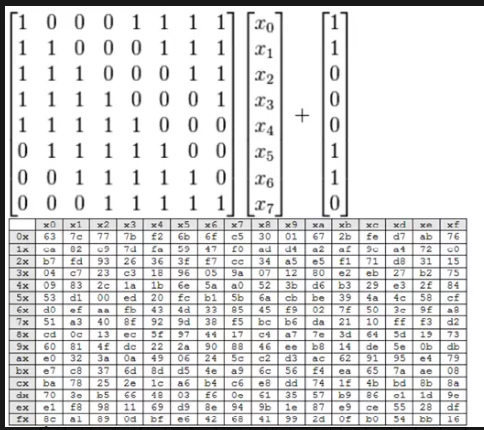
*Hình 7 Sơ đồ mã hóa AES 128 bits*

#### 2.2.1.2.Xây dựng thuật toán

* Xây dựng bảng S-Box

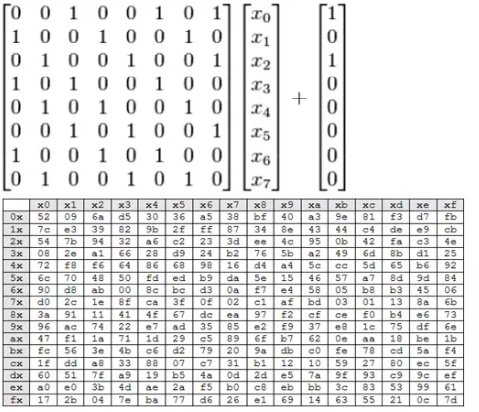
Bảng S-box thuận được sinh ra bằng việc xác định nghịch đảo cho một giá trị nhất định trên GF(28) = GF(2)[x] / (x8+x4+x3+x+1) (trường hữu hạn Rijindael). Giá trị 0 không có nghịch đảo thì được ánh xạ với 0. Những nghịch đảo được chuyển đổi thông qua phép biến đổi affine.

Công thức tính các giá trị bảng S-box và bảng S- box tương ứng: 



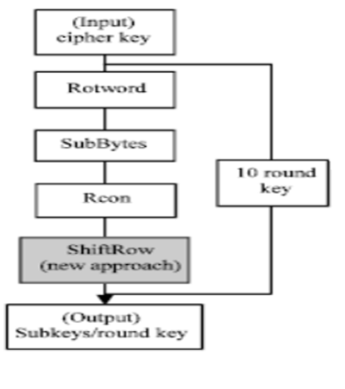
*Hình 8 bảng giá trị S-Box*

S-box nghịch đảo chỉ đơn giản là S-box chạy ngược. Nó được tính bằng phép biến đổi affine nghịch đảo các giá trị đầu vào. Phép biến đổi affine nghịch đảo được biểu diễn như sau:

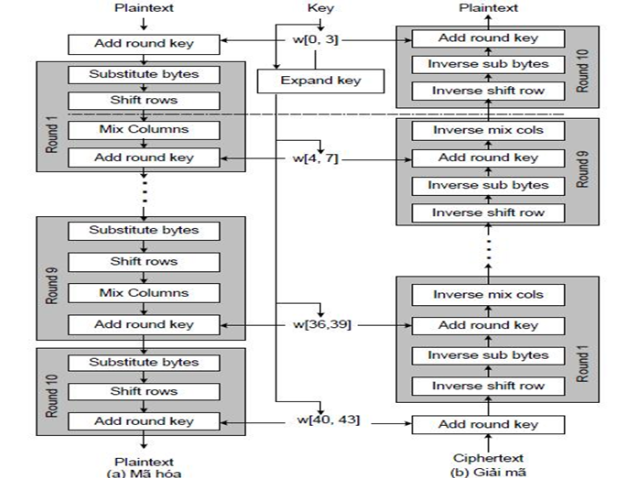


*Hình 9 Bảng giá trị S-box nghịch đảo*

* Giải thuật sinh khóa phụ
* Rotword: quay trái 8 bít
* SubBytes
* Rcon: tính giá trị Rcon(i) Trong đó :
* Rcon(i) = x(i-1) mod (x8 + x4 + x3 + x + 1).
* ShiftRow



* Quá trình mã hóa.



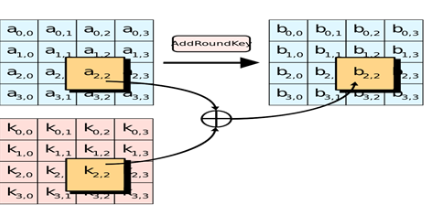
*Hình 10 Quá trình mã hóa*

1. Hàm AddRoundKey.

* Được áp dụng từ vòng lặp thứ 1 tới vòng lặp Nr
* Trong biến đổi Addroundkey(), một khóa vòng được cộng với state bằng một phép XOR theo từng bit đơn giản.
* Mỗi khóa vòng gồm có 4 từ (128 bit) được lấy từ lịch trình khóa. 4 từ đó được cộng vào mỗi cột của state, sao cho:

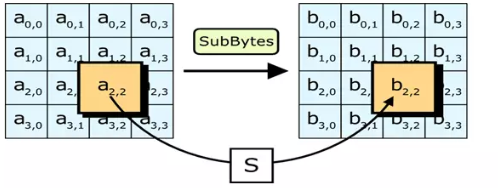
[S’0,c, S’1,c, S’2,c, S’3,c ] = [S0,c, S1,c, S2,c, S3,c ] ⊕ [W(4\*i + c)]

Với 0 <= c < 4.



1. Hàm SubBytes.

Biến đổi SubBytes() thay thế mỗi byte riêng rẽ của state Sr,c bằng một giá trị mới S’ r,c sử dụng bảng thay thế (S - box) được xây dựng ở trên.



1. Hàm ShiftRow.

* Trong biến đổi ShiftRows(), các byte trong ba hàng cuối cùng của trạng thái được dịch vòng đi các số byte khác nhau (độ lệch). Cụ thể :

S’r,c = Sr,(c + shift ( r, Nb)) mod Nb (Nb = 4)

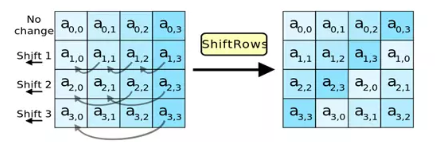
* Trong đó giá trị dịch shift (r, Nb) phụ thuộc vào số hàng r như sau:

Shift(1,4) = 1, shift(2,4) = 2, shift(3,4) = 3.

* Hàng đầu tiên không bị dịch, ba hàng còn lại bị dịch tương ứng:

Hàng thứ 1 giữ nguyên.

* Hàng thứ 2 dịch vòng trái 1 lần.
* Hàng thứ 3 dịch vòng trái 2 lần.
* Hàng thứ 4 dịch vòng trái 3 lần.



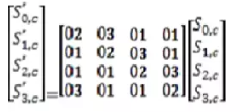
1. Hàm MixColumns.

* Biến đổi MixColumns() tính toán trên từng cột của state. Các cột được coi như là đa thức trong trường GF(28) và nhân với một đa thức a(x) với:

a(x) = (03)x^3 +(01)x^2 +(01)x + (02)

* Biến đổi này có thể được trình bày như phép nhân một ma trận, mà mỗi byte được hiểu như là một phần tử trong trường GF(28): s’(x) = a(x) ⊗

Mô tả bằng ma trận như sau :



* Giải mã

Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm sử dụng là 4 hàm ngược của quá trình mã hóa.

**2.2.1.3. Ưu , nhược điểm của hệ mã hóa AES :**

**\* Ưu điểm** :

Thiết kế và độ dài của thuật toán AES (128, 192, 256 bits) là đủ an toàn để bảo vệ các thông tin được xếp loại tối mật (secret) . Các phiên bản thực hiện AES nhằm mục đích bảo vệ hệ thống an ninh hay thng tin quốc gia phải được NSA kiểm định .

Vào tháng 6 năm 2003 , chính phủ Hoa Kì tuyên bố AES có thể sử dụng cho thông tin mật .

Vào thời điểm năm 2006 , dạng tấn công duy nhất thành công lên AES là tấn công kênh bên (side channel attack) . Tấn công kênh bên là tấn công trực tiếp vào thuật toán mã hóa , tấn công lên các hệ thống bảo vệ thuật toán có sơ hở làm lộ dữ liệu .

AES có mô hình toán học đơn giản , cấu trúc rõ ràng đơn giản .

**\* Nhược điểm** :

Về an ninh của AES , các nhà khoa học đánh giá là chưa cao . Họ cho rằng ranh giới giữa số chu kì của thuật toán và số chu kì bị phá vỡ quá nhỏ . Nếu các kĩ thuật tấn công dược cải thiện thì AES có thể bị phá vỡ .

Một vấn đề nữa là cấu trúc toán học của AES . AES có mô hình toán học khá đơn giản . Tuy điều này chưa dẫn tới mối nguy hiểm nào nhưng một số nhà ngiên cứu sợ rằng sẽ có người lợi dụng cấu trúc này trong tương lai .

**2.2.1.4 . Các dạng tấn công vào AES và phương pháp phòng chống.**

**a, Side-channel attack**

Side Channels (Kênh kề) được định nghĩa là các kênh đầu ra không mong muốn từ một hệ thống.

- Tấn công kênh bên hay còn gọi là Tấn công kênh kề là loại tấn công dễ thực hiện trong các loại tấn công mạnh chống lại quá trình triển khai mã hóa, và mục tiêu của loại tấn công này là phân tích các nguyên tố, các giao thức, modul, và các thiết bị trong mỗi hệ thống.

- Phân loại :

+ Tấn công thời gian.

+ Tấn công dựa vào lỗi.

+ Tấn công phân tích năng lượng.

+ Tấn công phân tích điện từ.

**b,Known attacks**

- Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là tấn công XSL và chỉ ra điểm yếu tiềm tàng của AES.

- Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có sai lầm trong tính toán. Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn để ngỏ và cho tới nay thì tấn công XSL vẫn chỉ là suy đoán.

**c,Các phương pháp phòng chống**

-*Phương pháp 1: Mã hóa cực mạnh*

Sử dụng các biện pháp để tăng tính bảo mật của các thuật toán mã hóa.

-*Phương pháp 2: Bảo vệ dữ liệu theo phương pháp vật lý*

Nếu một kẻ tấn công không thể tiếp cận vật lý với dữ liệu, dĩ nhiên khả năng đánh cắp khóa mã hóa sẽ khó khăn hơn. Vì vậy, trước những cuộc tấn công qua âm thanh tiềm tàng, bạn có thể sử dụng các giải pháp bảo vệ vật lý như đặt laptop vào các hộp cách ly âm thanh, không để ai lại gần máy tính khi đang giải mã dữ liệu hoặc sử dụng các nguồn âm thanh băng rộng tần số đủ cao để gây nhiễu.

- *Phương pháp 3: Kết hợp cả 2 cách trên*.

### 2.2.2. Thuật toán Euclid

Giải thuật Euclid mở rộng được sử dụng để giải một phương trình vô định nguyên (còn được gọi laf phương trình Đi-ô-phăng) có dạng: ax + by = c

Trong đó a, b, c là các hệ số nguyên, x, y là các ẩn nhận giá trị nguyên. Điều kiện cần và đủ để phương trình này có nghiệm (nguyên) là  UCLN(a,b) là ước của c . Khẳng định này dựa trên một mệnh đề sau:

*Nếu* d = UCLN(a,b)*thì tồn tại các số nguyên x,y  sao cho ax + by = d*

Cơ sở lý thuyết của giải thuật

Giải thuật Euclid mở rộng kết hợp quá trình tìm ƯCLN *(a, b)* trong thuật toán Euclid với việc tìm một cặp số x, y thoả mãn phương trình Đi-ô-phăng. Giả sử cho hai số tự nhiên a, b, ngoài ra a>b>0. Đặt r0 = a , r1 = b, chia cho r0  được số dư r2 và thương số nguyên q1 . Nếu r2 = 0 thì dừng, nếu r2 khác không, chia r1  cho r2  được số dư r3,...Vì dãy các ri là giảm thực sự nên sau hữu hạn bước ta được số dư rm+2 = 0 .

r0 = q1 \* r1 + r2 , 0<r2<r\_{1}};

r1 = q2 \* r2 + r3, 0<r3<r2, 0< r3 < r2;

....;

rm-1 = qm\* rm + rm+1, 0 < rm+1<rm

rm = qm+1 \* rm+1

trong đó số dư cuối cùng khác 0 là  rm+1 = d. Bài toán đặt ra là tìm x, y sao cho

a \* x + b\* y = rm+1 (=d)

Để làm điều này, ta tìm x, y theo công thức truy hồi, nghĩa là sẽ tìm xi và yi sao cho:

a \* xi + b\* yi = rm+1 (=d)  với i = 0,1,…

Ta có

a\*1+b\*0=a=r0 và a\*0+b\*1=b=r1 nghĩa là x0=1, x1=0 và y0=0, y1=1. (1)

Tổng quát, giả sử có

a\*xi+b\*y\_{i}=ri với i=0,1...

a\*xi+1+ b\*yi+1 = ri+1 với i=0,1...

Khi đó từ

ri = qi+1 \* ri+1 + ri+2

suy ra

ri - qi+1 \* ri+1 = ri+2

(a \* xi + b\* yi) - qi+1 \* (a\*xi+1+ b\*yi+1) = ri+2

a \* (xi - qi+1 \*xi+1) + b \* (yi - qi+1 \* yi+1) = ri+2

từ đó, có thể chọn

xi+2  = xi - qi+1 \*xi+1 (2)

yi+2 = yi - qi+1 \* yi+1 (3)

Khi i = m -1 ta có được xm+1 và ym+1 . Các công thức (1), (2), (3) là công thức truy hồi để tính x, y.

**Giải thuật**

{Thuật toán Euclide: a, b không đồng thời bằng 0, trả về gcd(a, b)}

function gcd(a, b);

begin

**while** b ≠ 0 **do**

begin

r:= a mod b; a:= b; b:= r;

end;

Result:= a;

end;

{Thuật toán Euclide mở rộng: a, b không đồng thời bằng 0, trả về cặp (x, y) sao cho a \* x + b \* y = gcd(a, b)

Về tư tưởng là ghép quá trình tính cặp số (x, y) vào trong vòng lặp chính của thuật toán Euclide.}

function Extended\_gcd(a, b);

begin

(xa, ya):= (1, 0);

(xb, yb):= (0, 1);

**while** b ≠ 0 **do**

begin

q:= a div b;

r:= a mod b; a:= b; b:= r; *//Đoạn này giống thuật toán Euclide.*

(xr, yr):= (xa, ya) - q \* (xb, yb); *//Hiểu là: (xr, yr):= (xa, ya) "mod" (xb, yb);*

(xa, ya):= (xb, yb);

(xb, yb):= (xr, yr);

end;

Result:= (xa, ya);

end;

Giải thuật sau chỉ thực hiện với các số nguyên a>b>0, biểu diễn bằng giải mã:

**Sub** Euclid\_Extended(a,b)

**Dim** x0, x, y,y1 **AsSingle**

x0=1: x1=0: y0=0: y1=1

**While** b>0

r= a **mod** b

**if** r=0 **thenExitWhile**

q= a / b

x= x0-x1\*q

y= y0-y1\*q

a=b

b=r

x0=x1

x1=x

y0=y1

y1=y

**Wend**

**Me**.Print d:=b, x, y

code

**EndSub**

**Ví dụ:** Với a=29, b=8, giải thuật trải qua các bước như sau:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bước |  |  |  |  |  |  |  |  |  |  |
| 0 | 29 | 8 | 5 | 3 | 1 | 0 | 1 | 0 | 1 | -3 |
| 1 | 8 | 5 | 3 | 1 | 0 | 1 | -1 | 1 | -3 | 4 |
| 2 | 5 | 3 | 2 | 1 | 1 | -1 | 2 | -3 | 4 | -7 |
| 3 | 3 | 2 | 1 | 1 | -1 | 2 | -3 | 4 | -7 | 11 |
| 4 | 2 | 1 | 0 | 2 |  |  |  |  |  |  |

Kết quả thuật toán cho đồng thời d = UCLN(29,8) = 1 và x = -3, y=11  .  
Dễ dàng kiểm tra hệ thức  29\* (-3) + 8\*11 = 1

Áp dụng giải thuật Euclid mở rộng tìm số nghịch đảo trong vành  Zm

**Số nghịch đảo trong vành** Zm

Trong lý thuyết số, vành Zm được định nghĩa là vành thương của Z với quan hệ đồng dư theo modulo m (là quan hệ tương đương) mà các phần tử của nó là các lớp đồng dư theo modulo m (m là số nguyên dương lớn hơn 1). Ta cũng có thể xét Zm  chỉ với các đại diện của nó. Khi đó

Zm = {0,1,..,m-1}

Phép cộng và nhân trong Zm  là phép toán thông thường được rút gọn theo modulo m:

a + b = (a + b) mod m

a \* b = (a \* b) mod m

Phần tử a của Zm  được gọi là khả nghịch trong Zm  hay khả nghịch theo modulo m nếu tồn tại phần tử a' trong Zm sao cho a\*a'=1 trong Zm  hay a\*a'=1 (mod m) . Khi đó a' được gọi là nghịch đảo modulo m của a. Trong lý thuyết số đã chứng minh rằng, số a là khả nghịch theo modulo m khi và chỉ khi ƯCLN của a và m bằng 1.  
Khi đó tồn tại các số nguyên x, y sao cho:

m\*x + a\*y = 1

Đẳng thức này lại chỉ ra y là nghịch đảo của a theo modulo m. Do đó có thể tìm được phần tử nghịch đảo của a theo modulo m nhờ thuật toán Euclid mở rộng khi chia m cho a.

**Giải thuật**

*//a, m > 0. Trả về a^-1 mod m, gcd(a, m) phải bằng 1, chú ý là ta không cần quan tâm y khi giải pt diophante a \* x + m \* y = 1*

**function** ModuloInverse(a, m);

**begin**

xa:= 1; xm:= 0;

**while** m ≠ 0 **do**

**begin**

q:= a **div** m;

xr:= xa - q \* xm;

xa:= xm;

xm:= xr;

r:= a **mod** m;

a:= m;

m:= r;

**end**;

Result:= xa;

**end**;

Giải thuật sau chỉ thực hiện với các số nguyên m>a>0, biểu diễn bằng giã mã:

**Procedure Euclid\_Extended (a,m)**

*int*, y0:=0,y1:=1;

**While** a>0 **do** {

r:= m mod a

if r=0 then Break

q:= m div a

y:= y0-y1\*q

y0:=y1

y1:=y

m:=a

a:=r

}

**If** a>1 **Then Return** "A không khả nghịch theo mođun m"

**else Return** " Nghịch đảo modulo m của a là y"

**Ví dụ**: Tìm số nghịch đảo (nếu có) của 30 theo môđun 101

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bước i | **m** | **A** | **r** | **q** | **y0** | **y1** | **y** |
| 0 | **101** | **30** | 11 | 3 | 0 | 1 | -3 |
| 1 | 30 | 11 | 8 | 2 | 1 | -3 | 7 |
| 2 | 11 | 8 | 3 | 1 | -3 | 7 | -10 |
| 3 | 8 | 3 | 2 | 2 | 7 | -10 | 27 |
| 4 | 3 | 2 | 1 | 1 | -10 | 27 | **-37** |
| 5 | 2 | 1 | 0 | . | . | . | . |

Kết quả tính toán trong bảng cho ta -37 . Lấy số đối của 37 theo mođun 101 được 64. Vậy 30-1 mod 101 = 64.

**Ứng dụng**

Số nghịch đảo theo môđun được ứng dụng nhiều trong việc giải phương trình đồng dư, trong lý thuyết mật mã.

### 2.2.3. Định lý Fermat

“Nếu là một số nguyên dương và là một số nguyên tố thì A picture containing black, darkness

Description automatically generated“

Chứng minh 1. Sử dụng phương pháp quy nạp theo

Với = 1 thì mệnh đề luôn đúng.

Giả sử mệnh đề đúng đến a tức là p | ap - a .

Ta sẽ chứng minh mệnh đề đúng đến a + 1. Thật vậy:

(a+ 1)p − (a+ 1) = (ap - a ) + kp ak

Sử dụng p| Ckp với 1 ≤ k ≤p− 1 và giả thiết quy nạp ta suy ra

p |(a+ 1)p − ( a+ 1). Khi đó (a+ 1)p≡ (a+ 1) (mod p) .

Vậy ta hoàn tất chứng minh.

Chứng minh 2. Giả sử rằng gcd(a, p ) =1 và cần chứng minh rằng ap-1-1≡ 1(mod p).

Xét các số nguyên a, 2a , … , ( p− 1)a mà các số dư khi chia cho p phân biệt ( nếu không thì, với ia≡ ja(mod p) thì p|( i−j )a hay là p|i − j, dấu “=” xảy ra chỉ nếu i=j ).

Do đó a. (2a) … (p− 1)a ≡ 12 … ( p− 1)(mod p) .

Vì gcd(q, (p− 1)!) = 1 nên ta suy ra điều phải chứng minh.

**Lưu ý.** Định lý này có thể biết gọn dưới dạng: A picture containing black, darkness

Description automatically generated

### 2.2.4. Hàm số Euler

Cho số nguyên dương n, số lượng các số nguyên dương bé hơn n và nguyên tố cùng nhau với n được ký hiệu f(n) và gọi là hàm Euler.

Nhận xét: Nếu p là số nguyên tố, thì f(p) = p -1

Ví dụ: Tập các số nguyên không âm nhỏ hơn 7 là Z7 = {0, 1, 2, 3, 4, 5, 6}.

Do 7 là số nguyên tố, nên tập các số nguyên dương nhỏ hơn 7 và nguyên tố cùng nhau với 7 là Z7\* = {1, 2, 3, 4, 5, 6}.

Khi đó |Z| = f(p) = p - 1 = 7 - 1 = 6

\* Định lý về hàm Euler: Nếu n là tích của hai số nguyên tố p, q thì

f(n) = f(p) .f(q) = (p-1).(q-1)

\* Quan hệ Đồng dư

Khái niệm: Cho hai số nguyên a, b, m (m > 0). Ta nói rằng a và b “đồng dư” với nhau theo modulo m, nếu chia a và b cho m, ta nhận được cùng một số dư.

Ký hiệu: a º b (mod m).

Ví dụ: 17 º 5 (mod 3) vì chia 17 và 5 cho 3, được cùng số dư là 2.

Tính chất của quan hệ đồng dư:

- Với mọi số nguyên dương m ta có:

a ≡ a (mod m) với mọi a thuộc Z; (tính chất phản xạ)

a ≡ b (mod m) thì b ≡ a (mod m); (tính chất đối xứng)

a≡ b (mod m) và b≡c (mod m) thì a≡c (mod m); (tính chất bắc cầu)

- Tổng hay hiệu các đồng dư:

(a+b) (mod n) ≡ [(a mod n) + (b mod n)] (mod n)

(a-b) (mod n) ≡ [(a mod n) - (b mod n)] (mod n)

- Tích các đồng dư:

(a\*b) (mod n) ≡ [(a mod n) \* (b mod n)] (mod n)

\* Tập thặng dư thu gọn theo modulo

Khái niệm: Kí hiệu Z n = {0, 1, 2, ..., n-1} là tập các số nguyên không âm < n.

Với Z n và phép cộng (+) lập thành nhóm Cyclic có phần tử sinh là 1, phần tử trung lập e = 0, (Z n , +) gọi là nhóm cộng, đó là nhóm hữu hạn có cấp n.

Với phép (+) là phép cộng thông thường của các số nguyên.

Kí hiệu Z\*n = {x Î Z n , x là nguyên tố cùng nhau với n}. Tức là x phải ¹ 0.

Z\*n được gọi là Tập thặng dư thu gọn theo mod n, có số phần tử là f(n). Z\*n với phép nhân mod n lập thành một nhóm (nhóm nhân), phần tử trung lập e = 1. Tổng quát (Z\*n, phép nhân mod n) không phải là nhóm Cyclic. Nhóm nhân Z\*n là Cyclic chỉ khi n có dạng: 2, 4, p k hay 2p k với p là nguyên tố lẻ.

Ví dụ:Cho n = 21, Z\*n = {1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20}.

\* Độ phức tạp của thuật toán

Bài toán được diễn đạt bằng hai phần:

Đầu vào: Các dữ liệu vào của bài toán.

Đầu ra: Các dữ liệu ra của bài toán (kết quả).

Khái niệm Thuật toán: “Thuật toán” được hiểu đơn giản là cách thức để giải một bài toán. Cũng có thể được hiểu bằng hai quan niệm: trực giác hay hình thức như sau:

Một cách trực giác, Thuật toán được hiểu là một dãy hữu hạn các quy tắc (chỉ thị, mệnh lệnh) mô tả một quá trình tính toán, để từ dữ liệu đã cho đầu vào ta nhận được kết quả đầu ra của bài toán.

Một cách hình thức, người ta quan niệm thuật toán là một máy Turing. Thuật toán được chia thành hai loại: Đơn định và không đơn định. Chi phí thời gian của một quá trình tính toán là thời gian cần thiết để thực hiện một quá trình tính toán. Với thuật toán tựa Algol: Chi phí thời gian là số các phép tính cơ bản thực hiện trong quá trình tính toán.

Chi phí bộ nhớ của một quá trình tính toán là số ô nhớ cần thiết để thực hiện một quá trình tính toán. Gọi A là thuật toán, e là dữ liệu vào của bài toán đã được mã hóa bằng cách nào đó. Thuật toán A tính trên dữ liệu vào e phải trả một giá nhất định. Ta ký hiệu: tA(e) là giá thời gian và IA(e) là giá bộ nhớ.

### 2.2.5. Thuật toán Miller - Rabin:

#### \* Kiểm tra Miller-Rabin

Kiểm tra Miller-Rabin là một thuật toán xác suất để [kiểm tra tính nguyên tố](https://wiki.edu.vn/wiki/index.php?title=Ki%E1%BB%83m_tra_t%C3%ADnh_nguy%C3%AAn_t%E1%BB%91) cũng như các thuật toán kiểm tra tính nguyên tố: [Kiểm tra Fermat](https://wiki.edu.vn/wiki/index.php?title=Ki%E1%BB%83m_tra_Fermat) và [Kiểm tra Solovay-Strassen](https://wiki.edu.vn/wiki/index.php?title=Ki%E1%BB%83m_tra_Solovay-Strassen). Nó được đề xuất đầu tiên bởi [Gary L. Miller](https://wiki.edu.vn/wiki/index.php?title=Gary_L._Miller&action=edit&redlink=1) như một thuật toán tất định, dựa trên [giả thiết Riemann tổng quát](https://wiki.edu.vn/wiki/index.php?title=Gi%E1%BA%A3_thi%E1%BA%BFt_Riemann_t%E1%BB%95ng_qu%C3%A1t&action=edit&redlink=1); [Michael O. Rabin](https://wiki.edu.vn/wiki/index.php?title=Michael_O._Rabin&action=edit&redlink=1) đã sửa chữa nó thành một [thuật toán xác suất](https://wiki.edu.vn/wiki/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_x%C3%A1c_su%E1%BA%A5t&action=edit&redlink=1).

Khi sử dụng kiểm tra Miller-Rabin chúng ta căn cứ vào một mệnh đề Q(p, a) đúng với các số nguyên tố p và mọi số tự nhiên và kiểm tra xem chúng có đúng với số n muốn kiểm tra và một số được chọn ngẫu nhiên hay không? Nếu mệnh đề Q(n, a) không đúng, tất yếu n không phải là số nguyên tố, còn nếu Q(n,a) đúng, số n có thể là số nguyên tố với một xác suất nào đó. Khi tăng số lần thử, xác suất để n là số nguyên tố tăng lên.

\* Tiêu chuẩn kiểm tra Q(n, a):

\* Căn bậc hai của 1 trong

Trước hết là một bổ đề về căn bậc hai của đơn vị trong [trường hữu hạn](https://wiki.edu.vn/wiki/index.php?title=Tr%C6%B0%E1%BB%9Dng_h%E1%BB%AFu_h%E1%BA%A1n&action=edit&redlink=1), trong đó *p* là số nguyên tố. Chắc chắn rằng 1 và -1 luôn là các căn bậc hai của 1 theo mođun *p*. Chúng là hai căn bậc hai duy nhất của 1. Thật vậy, giả sử rằng *x* là một căn bậc hai của 1 theo mođun *p*. Khi đó:

Từ đó, x-1hoặc x+1là chia hết cho *p*.

##### \* Tiêu chuẩn Miler-Rabin:

Bây giờ giả sử *p* là một số nguyên tố lẻ, khi đó *p* - 1 là số chẵn và ta có thể viết *p* − 1 dưới dạng trong đó *s* là một số tự nhiên >=1 và *m* là số lẻ - Điều này nghĩa là ta rút hết các thừa số 2 khỏi *p* − 1. Lấy số *a* bất kỳ trong tập {*1,2,..,p-1*}. Xét dãy số với k=0,1,2,...,s. Khi đó , với k=1,2,...,s và

Từ định lý Fermat nhỏ:

hay

hay

Do đó, hoặc hoặc

Nếu ta dừng lại, còn nếu ngược lại ta tiếp tục với .

Sau một số hữu hạn bước

· hoặc ta có một chỉ số k, sao cho ,

· hoặc tới k=0 ta vẫn có .

Ta có mệnh đề Q(p,a) như sau:

*Nếu p là số nguyên tố lẻ và p* - 1 = thì với mọi a: 0<a<p-1:

* *hoặc , với mọi k=0,1,2,...,s*
* *hoặc tồn tai k: sao cho* .

\* Số giả nguyên tố

·Theo định lý Fermat nhỏ, với số nguyên tố *p* ta có với mọi a {1,2,...,p-1}:

· **Định nghĩa**. Hợp số *n* thoả mãn với a nào đó được gọi là số giả nguyên tố Fermat cơ sở a.

· Số Carmichael: Hợp số *n* là số giả nguyên tố Fermat với mọi cơ sở a <math>\in</math> {1…n}, ƯCLN(a,n)=1 được gọi là số Carmichael.

· **Định nghĩa**: Hợp số *n* được gọi là số giả nguyên tố mạnh Fermat cơ sở *a* nếu nó thoả mãn mệnh đề Q(n,a).

\* Giải thuật kiểm tra Miller-Rabin

Miller-Rabin test

INPUT Số tự nhiên lẻ *n*.

OUPTUT NguyenTo: TRUE/FALSE

1. Phân tích <math>n-1 = 2^s \cdot m</math> trong đó s <math>\ge</math> 1 và m là số tự nhiên lẻ

2. Chọn ngẫu nhiên số tự nhiên a <math>\in</math> {2,...,n-1}.

3. Đặt <math>b=a^m \pmod n</math>

4. Nếu<math> b \equiv -1 \pmod n</math> thì trả về TRUE. Kết thúc.

5. Cho k chạy từ 0 đến s:

1. Nếu <math>b \equiv -1 \pmod n</math> thì trả về TRUE. Kết thúc.

2. Thay b:=<math> b^2 \pmod n</math>.

6. Trả lời FALSE. Kết thúc.

Xác suất trả lời sai

· *Định lý*: nếu n là hợp số dương lẻ thì trong các số a <math>\in</math> {2...n-1} tồn tại không quá <math>\frac {n-1} {4}</math> cơ sở a để n là số giả nguyên tố mạnh Fermat.

· Gọi A là biến cố "*Số n là hợp số*". B là biến cố "*Kiểm tra Miller-Rabin trả lời n là số nguyên tố*". Khi đó xác suất sai của kiểm tra này là xác suất để số n là hợp số trong khi thuật toán cho câu trả lời TRUE, nghĩa là xác suất điều kiện P(A|B).

Theo định lý trên nếu n là hợp số thì khả năng kiểm tra này trả lời TRUE xảy ra với xác suất không vượt quá <math>\frac 1 4</math>, nghĩa là P(B|A)<math>\le \frac 1 4</math>. Tuy nhiên để tính xác suất sai của kiểm tra Miller-Rbin cần tính xác suất diều kiện P(A|B). Dựa trên định lý về ước lượng số các số nguyên tố ta đưa ra ước lượng

<math></math>

Theo định lý Bayes trong lý thuyết xác suất ta có công thức để tính xác suất sai của kiểm tra Miller-Rabin là:

P(A|B) = <math>\frac {P(B|A)\*P(A)}{P(B)}</math>

=<math>\frac {P(B|A)\*P(A)}{P(B|A)P(A)+P(B|\overline A)\*P(\overline A)}</math>

Trong công thức này P(A) đã biết ở trên, P(B|A) <math>\le \frac 1 4</math>, còn <math>P(B|\overline A)</math>= 1 vì khi n là số nguyên tố thì chắc chắn mệnh đề Q(n,a) là đúng và <math>P(\overline A)= 1- P(A)= \frac 2 {\ln n}</math>. Từ đó

P(A|B)=<math>\frac {P(B|A)\*P(A)}{P(B|A)P(A)+P(\overline A)}</math>

<math>P(A|B)\approx \frac {P(B|A)\*(\ln n-2)}{P(B|A)\*(\ln n-2)+2}</math>

(Tham khảo: Douglas R. Stisnon. Cryptography Theory and Practice.)

Kiểm tra Miller-Rabin lặp

Theo công thức tính xác suất sai trên đây, với n lớn (cỡ 130 chữ số thập phân), nếu thực hiện phép thử Miller-Rabin chỉ một lần, xác suất sai là khá lớn, tới trên 90%. Để giảm xác suất sai, ta lặp lại phép thử k lần với k số ngẫu nhiên a khác nhau, nếu n vượt qua 50 lần thử thì P(B|A)<math> \le \frac 1 {4^k}</math>, khi thay vào công thức với 50 lần thử nếu cả 50 lần, phép thử đều *"dương tính"* thì xác suất sai giảm xưống chỉ còn là một số rất nhỏ không vượt quá <math>9 \cdot 10^{-29}</math>.

Sử dụng Miller trong thực tếmột cách tổng quát thì thuật toán này chưa được chứng minh là đúng. Tính đúng đắn của nó phụ thuộc vào giả thuyết Reimann.

Tuy nhiên, với nn nhỏ (n<3,317,044,064,679,887,385,961,981n<3,317,044,064,679,887,385,961,981) thì thuật toán đã được kiểm tra và chứng minh là đúng. Ngoài ra, ta cũng không cần kiểm tra hết tất các các số nguyên aa như ở trên, mà chỉ cần dùng một vài số là đủ.

Bảng sau được sử dụng để kiểm tra số nguyên tố trong thực tế:

|  |  |
| --- | --- |
| **Giới hạn của** nn | **Các số** aa **cần dùng** |
| n<2,047n<2,047 | 2 |
| n<1,373,653n<1,373,653 | 2, 3 |
| n<9,080,191n<9,080,191 | 31, 73 |
| n<4,759,123,141n<4,759,123,141 | 2, 7, 61 |
| n<1,122,004,669,633n<1,122,004,669,633 | 2, 13, 23, 1662803 |
| n<2,152,302,898,747n<2,152,302,898,747 | 2, 3, 5, 7, 11 |
| n<3,474,749,660,383n<3,474,749,660,383 | 2, 3, 5, 7, 11, 13 |
| n<341,550,071,728,321n<341,550,071,728,321 | 2, 3, 5, 7, 11, 13, 17 |
| n<3,825,123,056,546,413,051n<3,825,123,056,546,413,051 | 2, 3, 5, 7, 11, 13, 17, 19, 23 |
| n<318,665,857,834,031,151,167,461n<318,665,857,834,031,151,167,461 | 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37 |
| n<3,317,044,064,679,887,385,961,981n<3,317,044,064,679,887,385,961,981 | 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41 |

Xem bảng trên, ta thấy chỉ cần dùng 3 số thử: 2, 7, 61 là đủ để kiểm tra tính nguyên tố của mọi số nguyên 32-bit.

Để hiểu rõ hơn cách dùng, bạn đọc xem thêm phần dưới.

Cài đặt thuật toán

Trong phần này mình sẽ dùng Miller để giải 2 bài [PNUMBER](http://vn.spoj.com/problems/PNUMBER/) và [PRIME1](http://www.spoj.com/problems/PRIME1/).

Thuật toán gồm các giai đoạn chính:

* Phân tích n−1n−1 thành dạng 2s×d2s×d.
* Thử lần lượt các số aa trong bảng ở trên.
* Khi thử, ta cần dùng hàm lũy thừa nhanh để tính admodnadmodn.

Ta sẽ lần lượt cài đặt các hàm cơ bản trên. Trước hết, ta khai báo các thư viện và kiểu dữ liệu cần sử dụng:

#include <tuple>

#include <iostream>

using namespace std;

typedef unsigned long long ll;

Hàm phân tích thành dạng 2s×d2s×d:

pair<ll, ll> factor(ll n) {

ll s = 0;

while ((n & 1) == 0) {

s++;

n >>= 1;

}

return {s, n};

}

Hàm tính admodnadmodn:

ll pow(ll a, ll d, ll n) {

ll result = 1;

a = a % n;

while (d > 0) {

if (d & 1) result = result \* a % n;

d >>= 1;

a = a \* a % n;

}

return result;

}

Hàm kiểm tra xem “nếu ad≢1(modn)ad≢1(modn) và (ad)2r≢−1(modn)(ad)2r≢−1(modn) với mọi rr từ 00 đến s−1s−1 thì nn **không phải là số nguyên tố**”:

// Trả về false nếu n không phải là số nguyên tố.

bool test\_a(ll s, ll d, ll n, ll a) {

if (n == a) return true;

ll p = pow(a, d, n);

if (p == 1) return true;

for (; s > 0; s--) {

if (p == n-1) return true;

p = p \* p % n;

}

return false;

}

Hàm cuối cùng: Thử kiểm tra nn với các aa khác nhau. Vì ta đang giải bài [PNUMBER](http://vn.spoj.com/problems/PNUMBER/), giới hạn số trong bài này là n<200000n<200000, tra bảng trên ta thấy chỉ cần kiểm tra với a=2a=2 và a=3a=3 là đủ:

// Trả về true nếu n là số nguyên tố,

// false nếu n là hợp số.

bool miller(ll n) {

if (n < 2) return false;

if ((n & 1) == 0) return n == 2;

ll s, d;

tie(s, d) = factor(n-1);

return test\_a(s, d, n, 2) && test\_a(s, d, n, 3);

}

Giới hạn của bài PRIME1 là n≤109n≤109, nên ta cần dùng a=2a=2, a=7a=7 và a=61a=61.

Xử lý tràn số

Bảng ở trên có thể kiểm tra đến các số lớn hơn 64-bit, tuy nhiên khi cài đặt, nếu không sử dụng kiểu số nguyên lớn thì ta chỉ có thể sử dụng thuật toán với các số 32-bit.

Lý do là thuật toán có dùng phép nhân. Khi nhân 2 số 32-bit kết quả phải được chứa trong số 64-bit, nhân 2 số 64-bit thì kết quả phải được chứa trong số 128-bit. Vì vậy, để không bị tràn số thì thuật toán chỉ có thể dùng với số 32-bit.

Để khắc phục điều này, ta có thể tự viết hàm để vừa thực hiện nhân vừa thực hiện mod.

Đây là code tính a×b mod n a×b mod n của Dana Jacobsen trong một [câu trả lời](https://www.quora.com/How-can-I-execute-A-*-B-mod-C-without-overflow-if-A-and-B-are-lesser-than-C) trên Quora:

#include <stdint.h>

uint64\_t mulmod(uint64\_t a, uint64\_t b, uint64\_t n) {

// if (a >= n) a %= n; /\* Careful attention from the caller \*/

// if (b >= n) b %= n; /\* should make these unnecessary. \*/

uint64\_t r = 0;

if ((a|b) < (1ULL << 32)) return (a\*b) % n;

if (a < b) { uint64\_t t = a; a = b; b = t; }

if (n <= (1ULL << 63)) {

while (b > 0) {

if (b & 1) { r += a; if (r >= n) r -= n; }

b >>= 1;

if (b) { a += a; if (a >= n) a -= n; }

}

} else {

while (b > 0) {

if (b & 1) r = ((n-r) > a) ? r+a : r+a-n; /\* r = (r + a) % n \*/

b >>= 1;

if (b) a = ((n-a) > a) ? a+a : a+a-n; /\* a = (a + a) % n \*/

}

}

return r;

}

Đoạn code trên hơi phức tạp, bạn đọc có thể đọc đoạn code sau để hiểu tư tưởng thuật toán:

uint64\_t mulmod(uint64\_t a, uint64\_t b, uint64\_t n) {

uint64\_t r = 0;

while (b > 0) {

if (b % 2 != 0) r = (r + a) % n;

a = a \* 2 % n;

b = b / 2;

}

return r;

}

Đoạn code ở dưới đơn giản hơn, tuy nhiên nó chạy chậm hơn và chỉ chạy đúng vớn n có tối đa 63 bit.

Thay thế các phép nhân trong code Miller ở phần trên bằng hàm mulmod, ta có thể sử dụng được Miller cho các số 64-bit.

Độ phức tạp

Độ phức tạp của hàm mulmod là O(logn)O(log⁡n).

Mỗi lần kiểm tra tốn O(s+logd)=O(logn)O(s+log⁡d)=O(log⁡n) phép nhân, kết hợp với phép ĐPT của phép nhân nữa là O(log2n)O(log2⁡n).

Vậy độ phức tạp của Miller là O(klog2n)O(klog2⁡n) với kk là số các số aa cần kiểm tra.

### 2.2.6. Định lý phần dư Trung Hoa

Trong nhiều trường hợp ta muốn tìm cách để tăng tốc độ tính toán Modulo. Các phép toán trên modulo các số nhỏ tính nhanh nhiều so với các số lớn. Chính vì vậy nếu số lớn phân tích được thành tích của các số nhỏ, từng cặp nguyên tố cùng nhau, thì ta sẽ có cách tính hiệu quả nhờ vào định lý Phần dư Trung hoa. Tính toán trên modulo của một tích các số mod M với M= m1m2.mk , trong đó GCD(m, m) = 1, với mọii khác j. Định lý phần dư Trung Hoa cho phép làm việc trên từng modulo m riêng biệt. Vì thời gian tính toán các phép toán trên modulo tỷ lệ với kích thước của số lấy modulo nên điều đó sẽ nhanh hơn tính toán trên toàn bộ M.

\*Có thể triển khai Định lý Trung Hoa theo một số cách như sau:

Tính toán theo modulo số lớn. Để tính A mod M, với M khá lớn và A là biểu thức số học nào đó. Trước hết ta cần tính tất cả a = A mod mị. Sau đó sử dụng công thức

A-(40) mod M

trong đó

M= M/m

4 = M;x(M' mod m) for Isisk

Ví dụ: Tính 17 mod 77. Áp dụng định lý phần dư Trung họa, ta coi A = 178, m = 7, m = 11. Khi đó M = 11, M2 = 7 và 11- mod 7 = 4 mod 7= 2, suy ra c = 11\*2 = 22 7' mod 11 = 8, suy ra C2 = 7\*8 = 56

ay = 178 mod 7 = (17 mod 7) mod 7 = 38 mod 7 = (32)4 mod 7 = 2 az= 178 mod 11 = (17 mod 11) mod 11 = 68 mod 11 =

= (6)\* mod 11 = 34 mod 11 = 4 Vây A = 178 mod 77 = (2\*22 + 4\*56) mod 77 = 268 mod 77 = 37 mod 37

\*Giải hệ phương trình modulo. Cho a = x mod mị, với GCDm, m) = 1, với mọi 1

khác j. Khi đó ta cũng áp dụng Định lý phần dư Trung Hoa để tìm x. Ví dụ. Cho x = 5 mod 7 và x = 6 mod 11. Tìm x.

Áp dụng định lý phần dư Trung hoa, ta tính: 7'mod 11 = 8 và 11 mod 7 = 2. Như vậy x = (5\*2\*11+6\*8\*7) mod (7\*11) = 61 mod 77.

Một trong những kết quả hữu ích nhất của lý thuyết số là định lý còn lại của Trung Hoa (CRT). Về bản chất (CRT) nói rằng: Có thể tìm lại các số nguyên trong một phạm vi nhất định từ dư lượng của chúng theo mod có một bộ các module tương đối quan trọng theo cặp.

Mười số nguyên trong Z10, đó là số nguyên từ 0 đến 9. Có thể được khôi phục lại từ 2 dư lượng mod 2 và mod 5 là các phần tử tương đối nguyên tố của 10. Nói các dư lượng tồn tại của một chữ số thập phân x là r2 = 0 và r5 = 3.

Đó là: x mod 2 = 0 và x mod 5 = 3. Vì vậy x là số nguyên dương trong Z10, chia cho 5 dư 3, suy ra x = 8

CRT có thể được thể hiện bằng nhiều cách. Dưới đây là một công thức hữu ích khác: M = 𝛱𝑘𝑚𝑖 trong đó: 𝑚𝑖 là cặp tương đối nguyên tố gcd(𝑚𝑖, 𝑚𝑗 )=1 với 1 ≤ i, j ≤ k và i ≠ j. Ta có thể biểu diễn bất kì số nguyên A trong ZM bởi 1 bộ k có các phần tử trong 𝑍𝑚𝑖 sử dụng tương ứng sau:

CRT có thể được thể hiện bằng nhiều cách. Dưới đây là một công thức hữu ích khác: M = 𝛱𝑘 trong đó: 𝑚𝑖 là cặp tương đối nguyên tố gcd(𝑚𝑖, 𝑚𝑗 )=1 với 1 ≤ i, j ≤ k và i ≠ j. Ta có thể biểu diễn bất kì số nguyên A trong ZM bởi 1 bộ k có các phần tử trong 𝑍𝑚𝑖 sử dụng tương ứng sau:

A ↔ (a1, a2, ..., ak)

Trong đó: A ϵ ZM ; 𝑎𝑖 ϵ 𝑍𝑚𝑖 và 𝑎𝑖 = A mod 𝑚𝑖 với 1 ≤ i ≤ k

- Hai khẳng định của CRT như sau:

1. Lập bản đồ của phương trình (2.11) là một sự tương thích một – một giữa ZM và sản phẩm Cartesia 𝑍𝑚1 × 𝑍𝑚2 × × 𝑍𝑚𝑘. Nghĩa là đối với mỗi số nguyên A sao cho 0 ≤ A ≤ M, có 1 bộ k (a1, a2, ..., ak) với 0 ≤ 𝑎𝑖≤ 𝑚𝑖 đại diên cho nó và đối với mỗi bộ (a1, a2, , ak) có 1 số nguyên độc nhất A trong ZM

2. Các hoạt động thực hiện trên các thành phần của ZM có thể được thực hiện tương đương trên bộ k tương ứng bằng cách thực hiện các hoạt động độc lập trong mỗi vị trí ngang nhau trong hệ thống thích hợp.

### 2.2.7. Hệ mã hóa công khai RSA.

* Tổng quan về hệ mã công khai

A picture containing text, screenshot, font, line

Description automatically generated

*Sơ đồ của hệ mã công khai*

Hệ mã công khai sử dụng hai khóa có quan hệ toán học với nhau, tức là một khóa này được hình thành từ khóa kia: Người muốn nhận bản mã (Alice) tạo ra một khóa mật (private key) và từ khóa mật tính ra khóa công khai (public key) với một thủ tục không phức tạp, còn việc tìm khóa mật khi biết khóa công khai là bài toán khó giải được. Khóa công khai sẽ đưa đến cho người gửi bản tin (Bob) qua kênh công cộng. Và bản tin được Bob mã hóa bằng khóa công cộng. Bản mã truyền đến Alice, và nó được giải mã bằng khóa mật.

* Giới thiệu chung về hệ mã khóa công khai RSA

- Thuật toán được Ron Rivest, Adi Shamir và Len Adleman (R.S.A) mô tả lần đầu tiên vào năm 1977.

- Trước đó, vào năm 1973, Clifford Cocks – một nhà toán học người Anh đã mô tả một thuật toán tương tự.

- Nhưng tại thời điểm đó thì thuật toán này không khả thi và chưa bao giờ được thực nghiệm.

Mã hóa RSA chính là một thuật toán hay còn gọi là hệ mã hóa bất đối xứng có phạm vi ứng dụng rộng rãi và phổ biến trong công tác mã hóa và công nghệ chữ ký điện tử. Trong hệ mã hóa này, public key có thể chia sẻ công khai cho tất cả mọi người. Hoạt động gửi và nhận sẽ được can thiệp bởi RSA vì bản thân nó chứa hai khóa là khóa công khai và khóa bí mật để làm 2 nhiệm vụ bất đối xứng là mã hóa và giải mã. Không giống với loại mã hóa có khóa đối xứng, loại khóa bí mật của RSA tuyệt đối không được truyền thông tin ra bên ngoài ngay cả khi có thiết bị nghe trộm thì nếu đối tượng xấu không có khóa bí mật cũng sẽ không thể giải mã được thông tin đó. Như vậy rõ ràng với 2 tính năng mã hóa và giải mã tối ưu đến tuyệt đối ở một phương trình bất đối xứng như vậy cho nên giái trị của RSA vô cùng lớn, nó sẽ được sử dụng ở hầu hết mọi trường hợp cần bảo mật thông tin.

RSA được sử dụng rộng rãi Thuật toán mã hóa RSA thỏa mãn 5 yêu cầu của một hệ mã hiện đại:

+ Hệ bảo mật cao

+ Thao tác nhanh

+ Dùng chung được

+ Có ứng dụng rộng rãi

+ Có thể dùng xác định chủ nhân (dùng làm chữ ký điện tử)

Vì vậy, RSA được sử dụng rộng rãi trong công tác mã hóa và công nghệ chữ ký điện tử: truyền dẫn quỹ điện tử chuyển đổi thư điện tử, giao dịch tiền điện tử, thương mại điện tử tài chính- ngân hàng...

* Mô tả hoạt động

- Thuật toán RSA có hai khóa:

+ Khóa công khai (Public key): được công bố rộng rãi cho mọi người và được dùng để mã hóa.

+ Khóa bí mật (Private key): những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng.

- Tạo khóa: Thực thi ở đầu người nhận

* Bước 1: Chọn ngẫu nhiên 2 số nguyên tố p, q
* Bước 2: Tính số làm modul: n=p\*q, ∅ₙ = (p-1)\*(q-1)
* Bước 3: Chọn ngẫu nhiên khóa mã hóa b (1<b<∅ₙ) sao cho GCD(b,∅ₙ)=1
* Bước 4: Giải phương trình sau để tìm khóa giải mã a:

a = b⁻¹ mod ∅ₙ (theo thuật toán Euclide mở rộng)

b\*a=1 mod ∅ₙ với 0 ≤ a ≤ ∅ₙ

* Bước 5: Kpub = {b ,n} , Kpr = {a, p ,q}

- Mã hóa:

y = eKpub(x) = xb mod n

x Zn = {0,1…..n-1}

- Giải mã:

x = dKpr(y) = ya mod n

* Độ an toàn mã hóa RSA

- Độ an toàn của hệ thống RSA dựa trên hai vấn đề: bài toán phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA.

- Vì vậy, muốn xây dựng hệ RSA an toàn thì n=p\*q phải là một số đủ lớn, để không có khả năng phân tích nó về mặt tính toán. Để đảm bảo an toàn nên chọn các số nguyên tố p và q từ 100 chữ số trở lên.

|  |  |
| --- | --- |
| Số các chữ số trong số được phân tích | Thời gian phân tích |
| 50 | 4 giờ |
| 75 | 104 giờ |
| 100 | 74 năm |
| 200 | 4000 năm |
| 300 | 500000 năm |
| 500 | 4 x 10 ^ 25 năm |

Bảng thời gian phân tích mã RSA

- Cách thức phân phối khóa công khai là một trong những yếu tố quyết định đối với độ an toàn của RSA.

- Vấn đề này nảy sinh ra một lỗ hổng gọi là Man-in-the-middle attack (tấn công vào giữa).

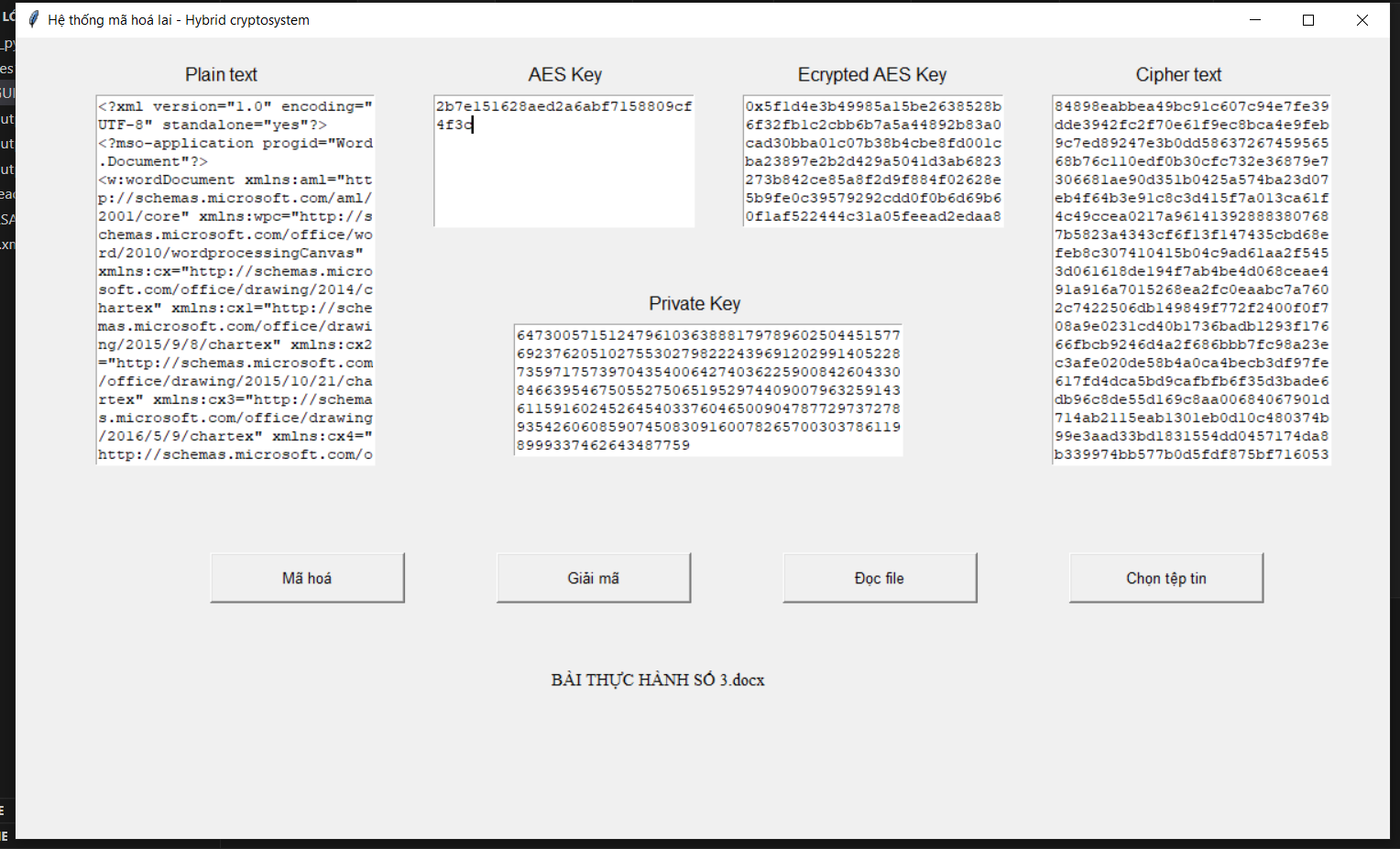
+ Khi A và B trao đổi thông tin thì C có thể gửi cho A một khóa bất kì để A tin rằng đó là khóa công khai của B gửi.

+ Sau đó C sẽ giải mã và đánh cắp được thông tin. Đồng thời mã hóa lại thông tin theo khóa công khai của B và gửi lại cho B.

+ Về nguyên tắc, cả A và B đều không phát hiện được sự can thiệp của C.

## 2.3.Thiết kế, cài đặt chương trình đề mô thuật toán

### 2.3.1.Giao diện chương trình đề mô



*Hình 13 : Demo chương trình bằng ngôn ngữ Python của Trần Đoàn Xuân Thành*

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 14 Demo chương trình bằng ngôn ngữ Java của Nguyễn Tiến Kiên*

Graphical user interface, application

Description automatically generated

*Hình 15 Demo chương trình bằng ngôn ngữ Python của Bùi Trung Kiên*

Graphical user interface, application

Description automatically generated

*Hình 16 Demo chương trình bằng ngôn ngữ Java của Nguyễn Viết Khánh*

### 2.3.2.Cài đặt và triển khai

#### 2.3.2.1.Giới thiệu công cụ

**Công Cụ:**

* Visual Studio Code và chạy chương trình trên Google Chorme (Thực hiện ngôn ngữ Javascript)
* IntelliJ IDEA 2021.2.2 (Thực hiện ngôn ngữ java)
* Visual Studio 2022, C# Windows form (.NET Framework 4.7.2)
* Visual Studio code (Thực hiện ngôn ngữ Dart)
* Pycharm (Thực hiện ngôn ngữ python)
* Cácthư viện base64, Crypto, tinyec, tkinter, … hỗ trợ.
* API: convertAPI

**Ngôn Ngữ:**

* + ***Dart***



*Hình 17 Ngôn ngữ lập trình Dart*

* Ngôn ngữ Dart là một ngôn ngữ lập trình được phát triển bởi Google. Nó được tạo ra để xây dựng ứng dụng web, ứng dụng di động, và cả ứng dụng máy tính trên nhiều nền tảng khác nhau. Dự án Dart ban đầu được giới thiệu vào năm 2011 và từ đó đã trải qua nhiều cải tiến và phát triển.
* Dart là một ngôn ngữ lập trình dễ đọc, dễ hiểu và dễ học. Nó kết hợp những ưu điểm từ nhiều ngôn ngữ khác nhau như JavaScript, Java và C#. Dart có cú pháp gần giống với nhiều ngôn ngữ lập trình thông dụng, giúp cho những lập trình viên đã quen thuộc với các ngôn ngữ khác có thể dễ dàng chuyển đổi sang Dart.
* Dart cung cấp một hệ sinh thái phong phú cho phát triển ứng dụng, bao gồm một bộ công cụ phát triển (SDK) mạnh mẽ và các thư viện hữu ích. Nó hỗ trợ viết mã một lần và chạy trên nhiều nền tảng khác nhau như web, di động (với Flutter), máy tính và máy chủ.
  + ***C#***



*Hình 18Ngôn ngữ lập trình C#*

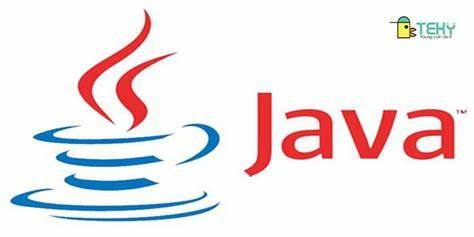
C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth. Là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java. C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

.NET framework có thể được sử dụng để tạo cả những ứng dụng dựa trên biểu mẫu (Form-based) và dựa trên Web (Web-based). Các web service cũng có thể được phát triển bằng cách sử dụng .NET framework.

***- Java***

Java được phát thành và phát triển từ năm 1995. Java được ứng dụng trên nhiều thiết bị như máy tính, điện thoại và phần cứng khác. Java là ngôn ngữ lập trình hướng đối tượng nên có đặc điểm chung với các ngôn ngữ khác như:

* + Tính trừu tượng: xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù.
  + Tính đa hình: Cho phép tác động khác nhau trên nhiều loại đối tượng khác nhau
  + Tính kế thừa: Cho phép các đối tượng chia sẽ, mở rộng các đặc tính sẵn có
  + Tính đóng gói: Che giấu việc thực thi những chi tiết của đối tượng đối với người sử dụng đối tượng ấy.



*Hình 19 Ngôn ngữ lập trình Java*

Java được ứng dụng nhiều trong đời sống thường ngày như:

Viết ứng dụng web: Java được sử dụng để xây dựng hệ thống web cần sự bảo mật cao, số lượng người dùng lớn

* + Viết ứng dụng mobile: Java được sử dụng để viết nên các ứng dụng, app hay game trên điện thoại
  + Viết ứng dụng desktop: Java sử dụng viết các ứng dụng trên máy tính. Chỉ cần viết một lần và sau đó có thể đưa chương trình lên windows để chạy mà không cần phải viết lại.

***-Python:***

Python là ngôn ngữ lập trình hướng đối tượng đơn giản, dễ học, mạnh mẽ, cấp cao. Python có cấu trúc cú pháp ít hơn các ngôn ngữ khác.

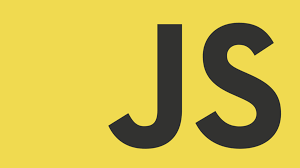


*Hình 20 Ngôn ngữ Python*

Các tính năng của Python bao gồm:

* Dễ học: Python có ít từ khóa, cấu trúc đơn giản và cú pháp được định nghĩa rõ ràng. Điều này cho phép người mới học tiếp cận ngôn ngữ một cách nhanh chóng.
* Dễ đọc: Mã Python được định nghĩa rõ ràng hơn và có thể nhìn thấy bằng mắt.
* Dễ bảo trì: Mã nguồn của Python khá dễ bảo trì.
* Một thư viện tiêu chuẩn rộng: Phần lớn thư viện của Python rất dễ đính kèm và đa nền tảng tương thích trên UNIX, Windows và Macintosh.
* Chế độ tương tác: Python có hỗ trợ cho chế độ tương tác cho phép kiểm tra tương tác và debug.
* Portable: Python có thể chạy trên nhiều nền tảng phần cứng khác nhau và có cùng giao diện trên tất cả các nền tảng.
* Có thể mở rộng: Bạn có thể thêm các module cấp thấp vào trình thông dịch Python. Các module này cho phép các lập trình viên thêm hoặc tùy chỉnh các công cụ của mình để hiệu quả hơn.
* Cơ sở dữ liệu: Python cung cấp phương thức giao tiếp cho tất cả các cơ sở dữ liệu.
* Lập trình GUI: Python hỗ trợ các ứng dụng GUI có thể được tạo và chuyển sang nhiều cuộc gọi hệ thống, thư viện và hệ thống cửa sổ, như Windows MFC, Macintosh và hệ thống X Window của Unix.
* Khả năng mở rộng: Python cung cấp cấu trúc và hỗ trợ tốt hơn cho các chương trình lớn hơn so với kịch bản lệnh shell.

**- Javascript:**

****

Hình 20: Ngôn ngữ Javascript

JavaScript (JS) là một ngôn ngữ lập trình phía client phổ biến và mạnh mẽ, được sử dụng để tạo ra các ứng dụng web tương tác và động. JS thường được nhúng trực tiếp vào mã HTML của trang web và thực thi trên trình duyệt của người dùng.

Dưới đây là vài điểm nổi bật về JavaScript:

* Cú pháp linh hoạt: JS có cú pháp dễ học và dễ hiểu. Nó dựa trên cú pháp giống C, do đó, những người đã quen thuộc với các ngôn ngữ như C++, Java hoặc C# có thể nhanh chóng chuyển đổi sang JS.
* Đa năng: JavaScript có thể được sử dụng để thực hiện nhiều tác vụ khác nhau trên các trang web, bao gồm xử lý sự kiện, tương tác với người dùng, thay đổi nội dung và giao diện của trang, gửi và nhận dữ liệu từ máy chủ, thực hiện kiểm tra hợp lệ dữ liệu và nhiều hơn nữa.
* Thư viện và framework phong phú: JavaScript có một hệ sinh thái phát triển mạnh mẽ với hàng ngàn thư viện và framework khác nhau như React, Angular, Vue.js, Node.js, Express.js, và nhiều thư viện hỗ trợ khác. Điều này giúp lập trình viên xây dựng ứng dụng phức tạp và mạnh mẽ một cách nhanh chóng và hiệu quả.
* Tương tác trực tiếp với DOM: JavaScript cho phép lập trình viên tương tác trực tiếp với Document Object Model (DOM) của trang web. Điều này cho phép thay đổi cấu trúc, nội dung và kiểu dáng của trang web một cách động, tạo ra hiệu ứng tương tác và cải thiện trải nghiệm người dùng.
* Hỗ trợ đa nền tảng: JavaScript không chỉ chạy trên trình duyệt web, mà còn được sử dụng để phát triển ứng dụng di động và desktop. Các công cụ như React Native và Electron cho phép lập trình viên sử dụng JavaScript để xây dựng ứng dụng di động và desktop đa nền tảng.

#### 2.3.2.2. Hướng dẫn cài đặt và chạy chương trình demo đã cài đặt

* **Chương trình viết bằng ngôn ngữ Dart**
* **Chương trình viết bằng ngôn ngữ C#**
* **Chương trình viết bằng ngôn ngữ Javascript**
* **Chương trình viết bằng ngôn ngữ Java**

1 . Mã hóa chuỗi :

- Điền Plaintext

- Điền khoá AES (128bit)

- Bấm Mã hóa để mã hóa

2 . Giải mã chuỗi :

- Điền Cipher text

- Điền khoá AES đã được mã hoá

- Điền khoá bí mật RSA

- Bấm Giải mã để giải mã

3. Mã hóa file (chỉ chấp nhận file có đuôi .txt, docx) :

- Chọn file cần mã hoá

- Đọc file cần mã hoá

- Điền khoá AES (128bit)

- Bấm Mã hóa để mã hóa

File sau khi mã hóa sẽ ở tệp file\_encrypted.txt và hiển thị ở ô Ciphertext

4.Giải mã file (chỉ chấp nhận file có đuôi .txt) :

- Chọn file cần giải mã

- Đọc file cần giải mã

- Điền khoá AES đã được mã hoá

- Điền khoá bí mật RSA (mỗi thành phần của khoá được ngăn cách bởi 1 kí tự line feed)

- Bấm Giải mã để giải mã

File sau khi giải mã sẽ tệp output.txt hoặc output.docx và hiển thị ở ô plaintext

* **Chương trình viết bằng ngôn ngữ Python**.

1 . Mã hóa chuỗi :

- Điền Plaintext

- Điền khoá AES (128bit)

- Bấm Mã hóa để mã hóa

2 . Giải mã chuỗi :

- Điền Cipher

- Điền khoá AES đã được mã hoá

- Điền khoá bí mật RSA (mỗi thành phần của khoá được ngăn cách bởi 1 kí tự line feed)

- Bấm Giải mã để giải mã

3. Mã hóa file (chỉ chấp nhận file có đuôi .txt,docx) :

- Chọn file cần mã hoá

- Đọc file cần mã hoá

- Điền khoá AES (128bit)

- Bấm Mã hóa để mã hóa

File sau khi mã hóa sẽ ở tệp output.txt và hiển thị ở ô Ciphertext

4.Giải mã file (chỉ chấp nhận file có đuôi .txt) :

- Chọn file cần giải mã

- Đọc file cần giải mã

- Điền khoá AES đã được mã hoá

- Điền khoá bí mật RSA (mỗi thành phần của khoá được ngăn cách bởi 1 kí tự line feed)

- Bấm Giải mã để giải mã

File sau khi giải mã sẽ tệp output.txt hoặc output.docx và hiển thị ở ô plaintext

### 2.3.3.Thực hiện bài toán

#### 2.3.3.1.Phân công công việc

|  |  |
| --- | --- |
| **Tên sinh viên** | **Tên công việc** |
| Trần Đoàn Xuân Thành | Tìm hiểu về ATBMTT, hệ mã hóa AES và RSA cài đặt  thực hiện chương trình theo ngôn ngữ Python |
| Nguyễn Tiến Duy | Tìm hiểu về ATBMTT, hệ mã hóa AES và RSA cài đặt  thực hiện chương trình theo ngôn ngữ Dart |
| Nguyễn Tiến Kiên | Tìm hiểu về ATBMTT, hệ mã hóa AES và RSA cài đặt  thực hiện chương trình theo ngôn ngữ Java |
| Phạm Quốc Tiến | Tìm hiểu về ATBMTT, hệ mã hóa AES và RSA cài đặt  thực hiện chương trình theo ngôn ngữ Javascript |
| Đào Văn Nhật | Tìm hiểu về ATBMTT, hệ mã hóa AES và RSA cài đặt  thực hiện chương trình theo ngôn ngữ C# |

**Trần Đoàn Xuân Thành - Phương pháp mã hóa AES, RSA để đảm bảo tính an toàn, toàn vẹn dữ liệu bằng ngôn ngữ Python**

* Tìm hiểu về An Toàn Bảo Mật Thông Tin.
* Tìm hiểm về hệ mã hóa AES, RSA
* Tìm hiểu ngôn ngữ Python
* Viết chương trình đề mô với ngôn ngữ Python

**Nguyễn Tiến Duy - Phương pháp mã hóa AES, RSA để đảm bảo tính an toàn, toàn vẹn dữ liệu bằng ngôn ngữ Dart**

* Tìm hiểu về An Toàn Bảo Mật Thông Tin.
* Tìm hiểm về hệ mã hóa AES, RSA
* Tìm hiểu ngôn ngữ Dart
* Viết chương trình đề mô với ngôn ngữ Dart

**Nguyễn Tiến Kiên - Phương pháp mã hóa AES, RSA để đảm bảo tính an toàn, toàn vẹn dữ liệu bằng ngôn ngữ Java**

* Tìm hiểu về An Toàn Bảo Mật Thông Tin.
* Tìm hiểm về hệ mã hóa AES, RSA
* Tìm hiểu ngôn ngữ Java
* Viết chương trình đề mô với ngôn ngữ Java

**Phạm Quốc Tiến - Phương pháp mã hóa AES, RSA để đảm bảo tính an toàn, toàn vẹn dữ liệu bằng ngôn ngữ Javascript**

* Tìm hiểu về An Toàn Bảo Mật Thông Tin.
* Tìm hiểm về hệ mã hóa AES, RSA
* Tìm hiểu ngôn ngữ Javascript
* Viết chương trình đề mô với ngôn ngữ Javascript

**Đào Văn Nhật - Phương pháp mã hóa AES, RSA để đảm bảo tính an toàn, toàn vẹn dữ liệu bằng ngôn ngữ C#**

* Tìm hiểu về An Toàn Bảo Mật Thông Tin.
* Tìm hiểm về hệ mã hóa AES, RSA
* Tìm hiểu ngôn ngữ C#
* Viết chương trình đề mô với ngôn ngữ C#

**Tài liệu tham khảo :**

1. Giáo trình

[1]. Giáo trình An toàn và bảo mật thông tin - Thư viện điện tử trường Đại học Công nghiệp Hà Nội

<http://thuvienso.haui.edu.vn/tailieuvn/doc/giao-trinh-an-toan-bao-mat-thong-tin-2381912.html>

2. Internet

[2]. Tìm hiểu thuật toán mã hóa khóa đối xứng AES

https://en.wikipedia.org/wiki/Advanced\_Encryption\_Standard

<https://viblo.asia/p/tim-hieu-thuat-toan-ma-hoa-khoa-doi-xung-aes-gAm5yxOqldb>

<https://viblo.asia/p/tim-hieu-thuat-toan-ma-hoa-khoa-doi-xung-aes-gAm5yxOqldb>

<https://doc.edu.vn/tai-lieu/de-tai-tim-hieu-dua-ra-uu-va-nhuoc-diem-cac-phien-ban-cua-cac-thuat-toan-des-triple-des-aes-7643/>

<https://tinhte.vn/thread/tim-hieu-chuan-ma-hoa-aes-duoc-my-ap-dung-trong-viec-ma-hoa-du-lieu.2871055/>

[3]. Tìm hiểu thuật toán mã hóa RSA

<https://vi.wikipedia.org/wiki/Ch%E1%BB%AF_k%C3%BD_s%E1%BB%91>

[https://vi.wikipedia.org/wiki/RSA\_(m%C3%A3\_h%C3%B3a)](https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)%20)

<https://www.educba.com/rsa-algorithm/>