Nhom3 Bao Cao BTL Atbmtt

Pháp luật đại cương (Trường Đại học Thăng Long)

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

BÀI TẬP LỚN

Môn: An toàn và bảo mật thông tin

TÌM HIỂU VỀ CHỮ KÝ ĐIỆN TỬ RSA VÀ VIẾT ỨNG DỤNG MINH HỌA

CBHD: ***ThS. Trần Phương Nhung***

Lớp: 20222IT6001007

Nhóm: 3

Thành viên nhóm:

1. **Nguyễn Đức Đạt - 2021600374**
2. **Nguyễn Hữu Đạt - 2021600814**
3. **Nguyễn Tuấn Đạt - 2021600039**
4. **Nguyễn Thế Đoàn - 2021600270**
5. **Phạm Đăng Đông - 2021603320**

Hà Nội – Năm 2023

LỜI CẢM ƠN

Nhóm 3 chúng em xin gửi lời cảm ơn chân thành tới cô Trần Phương Nhung. Cám ơn cô đã tạo điều kiện cho nhóm thực hiện đề tài này. Qua đó chúng em có thể dùng những kiến thức được học trên trường áp dụng vào dự án thực tế. Hơn nữa, chúng em đã học được thêm nhiều kiến thức mới cũng như phát triển thêm một số kỹ năng mềm như kỹ năng làm việc nhóm, quản lý thời gian… Cảm ơn cô đã tận tình chỉ bảo, hướng dẫn nhóm trong quá trình học tập và thực hiện đề tài.

Để hoàn thiện được đề tài này, nhóm chúng em đã cùng nhau thảo luận, nghiên cứu, áp dụng những kiến thức được học cũng như tìm hiểu thực tế. Với một khoảng thời gian chưa nhiều, nhưng chúng em đã nỗ lực bằng tất cả khả năng của mình để hoàn thành sản phẩm này, rất mong cô và các bạn có thể đóng góp thêm ý kiến để dề tài có thể hoàn thiện hơn nữa.

Chúng em xin trân thành cảm ơn!

LỜI MỞ ĐẦU

Với sự phát triển của mạng Internet hiện nay, công nghệ được ứng dụng trong hầu hết các lĩnh vực của đời sống. Bên cạnh những cách làm truyền thống cũng đã xuất hiện những công nghệ mới được áp dụng và đem lại hiệu quả đáng kể. Bên cạnh việc áp dụng công nghệ để đem lại hiệu quả cao hơn thì nhu cầu bảo mật thông tin được đặt lên hàng đầu. Để giải quyết vấn đề xác nhận chữ ký truyền thống trong các văn bản giao dịch, việc áp dụng công nghệ thông tin thay đổi và giúp tối ưu việc xử lý và bảo mật hơn. Cách giải quyết hiệu quả được đưa ra đó là áp dụng chữ ký điện tử vào công việc.

Đề tài "Tìm hiểu về chữ ký điện tử RSA và viết ứng dụng minh họa" sẽ tìm hiểu về vấn đề nêu trên và cài đặt chương trình minh họa.

Nội dung chính của bài báo cáo bao gồm 3 chương, trong đó:

Chương 1: Tổng quan

Chương 2: Kết quả nghiên cứu

Chương 3: Kiến thức lĩnh hội và bài học kinh nghiệm

MỤC LỤC

DANH MỤC HÌNH ẢNH

Chương 1. Tổng quan

1.1 Mục đích chọn đề tài

Ngày nay, với sự phát triển không ngừng của công nghệ thông tin, công nghệ được ứng dụng trong hầu hết các lĩnh vực của đời sống. Công nghệ đã và đang đóng vai trò vô cùng to lớn vào sự phát triển của mọi lĩnh vực. Vì vậy, công nghệ bảo mật thông tin hiện nay là rất quan trọng.

Ví dụ trong các ngân hàng hay các giao dịch điện tử việc bảo mật thông tin cá nhân của người dùng là vô cùng quan trọng. Cho nên nhóm 3 chúng em chọn đề tài nghiên cứu về chữ ký điện tử trong bảo mật thông tin kết hợp với hệ mã hóa RSA và ứng dụng chữ ký điện tử RSA trong bài báo cáo môn An toàn bảo mật thông tin này.

Đề tài này chúng ta cần nắm vững được cách mã hóa thông tin bằng hệ mã hóa RSA và cách áp dụng chữ ký điện tử trong an toàn và bảo mật thông tin.

1.2 Xác định nội dung nghiên cứu

Dựa trên những kiến thức tự tìm hiểu và học được trong học phần An toàn và bảo mật thông tin, nhóm 3 chúng em áp dụng những hiểu biết về mã hóa thông tin và bảo mật liên quan đến hệ mật mã RSA và những nội dung sau để hoàn thành bài báo cáo:

- Chữ ký điện tử, chữ ký điện tử RSA
- Thuật toán bình phương và nhân
- Hàm băm MD5
- Phương pháp mã hóa bất đối xứng

Các chương trình ứng dụng được nghiên cứu trong báo cáo của nhóm thực hiện việc xây dựng giao diện và thuật toán xử lý tạo chữ ký điện tử theo phương pháp của hệ mã RSA sử dụng để tạo chữ ký điện tử cho nội dung sử dụng chữ ký.

1.3 Tổng quan về chữ ký số

1.3.1 Khái niệm

Trong đời sống hàng ngày, chữ ký trên một văn bản là một minh chứng về "bản quyền" hoặc ít nhất cũng là sự tán đồng, thừa nhận các nội dung trong văn bản. Chẳng hạn như việc ký vào phiếu nhận tiền từ ngân hàng, hợp đồng mua bán, … Chữ ký viết tay được chính tay người ký nên không thể sao chụp được. Thông thường chữ ký viết tay trên văn bản được dùng để xác nhận người ký nó. Những yếu tố nào làm nên sức thuyết phục của nó? Ta có thể xem xét các yếu tố sau:

- Chữ ký là bằng chứng thể hiện người ký có chủ định khi ký văn bản

- Chữ ký thể hiện chủ quyền, nó làm cho người nhận văn bản biết rằng ai là người đã ký văn bản
- Văn bản đã ký không thể thay đổi được
- Chữ ký không thể giả mạo và cũng là thứ không thể chối bỏ

Trong đời sống bình thường, việc tạo ra một mô hình lý tưởng như trên là không hề dễ dàng vì việc ký trên văn bản giấy có thể giả mạo chữ ký, nhưng với khả năng kiểm định sát sao thì việc làm thay đổi không phải là dễ. Tuy nhiên trong thế giới máy tính thì vấn đề ký như trên gặp phải nhiều khó khăn: các dòng thông tin trên máy tính có thể thay đổi dễ dàng, hình ảnh chữ ký tay của một người cũng dễ dàng sao chép từ một văn bản này sang một văn bản khác, và việc thay đổi nội dung một văn bản điện tử cũng chẳng để lại dấu vết gì về phương diện tẩy, xóa.

Để có được những đặc tính như trên, giao thức ký trong thế giới điện tử cần phải có sự hỗ trợ của công nghệ mã hóa. Sơ đồ chữ ký số là phương pháp ký một thông báo được lưu dưới dạng điện tử. Giao thức cơ bản của chữ ký số dựa trên ý tưởng của Diffie và Hellman:

- Người ký văn bản bằng cách mã hóa nó với khóa bí mật của mình
- Người gửi chuyển văn bản cho người nhận
- Người nhận văn bản kiểm tra chữ ký bằng việc sử dụng chìa khóa công khái của người gửi để giải mã văn bản

⇨ **Khái niệm**: *Chữ ký số là mô hình sử dụng các kỹ thuật mã hóa mật mã để gắn với mỗi người sử dụng một cặp khóa công khai – bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật. Khóa công khai thường được phân phối thông qua chứng thực khóa công khai.*

1.3.2 Vị trí, vai trò của chữ ký số

Xu hướng quốc tế hóa và toàn cầu hóa đã và đang ảnh hưởng đến sự phát triển của thế giới. Việc trao đổi thông tin cũng từ đó yêu cầu nhanh gọn, chính xác và đặc biệt là phải an toàn. Việc trao đổi thông tin, chứng thực thông tin theo phong cách truyền thống làm giảm tốc độ, cũng như sự chính xác của thông tin. Những công việc đó mang tính chất thủ công gây ra sự chậm chễ và thiếu chính xác trong trao đổi.

Chính khó khăn đã nảy sinh sự phát triển mạnh mẽ của công nghệ thông tin và công nghệ mã hóa. Hiện nay ở tất cả các nước phát triển cũng như đang phát triển, mạng máy tính đang ngày càng đóng vai trò thiết yếu trong mọi lĩnh vực hoạt động của toàn xã hội và nhu cầu bảo mật thông tin đặt lên hàng đầu. Điển hình là việc mã hóa bảo mật các

thông tin số của doanh nghiệp, dùng chữ ký số xác thực email trao đổi thông tin, kiểm soát truy cập vào các sàn thương mại điện tử và các đơn đặt hàng, ngân hàng điện tử, mua sắm trực tuyến… mà vai trò chủ yếu là chữ kí số điện tử.

Trên thực tế chữ kí số không chỉ được thực hiện cho các giao dịch điện tử trên mạng internet mà còn qua hệ thống mạng viễn thông di động. Đặc biệt, hiện nay nhiều nước trên thế giới không chỉ triển khai ứng dụng chữ ký số trên mạng máy tính mà còn áp dụng trên mạng điện thoại di động để thực hiện các giao dịch điện tử. Hướng đi này giúp đẩy nhanh giao dịch, đơn giản hóa mua sắm trực tuyến và giúp người dùng có thể truy cập mọi lúc mọi nơi.

Sự ra đời của chữ kí số khẳng định được lợi ích to lớn về chiến lược và kinh tế, đồng thời các vấn đề liên quan đến chữ kí số cũng là những chủ đề quan trọng nhất của mật mã học.

1.3.3 Sơ đồ tổng quan của chữ ký số

- Chữ ký số điện tử bao gồm 3 thành phần: thuật toán tạo khóa, hàm tạo chữ ký và hàm kiểm tra chữ ký.
- Hàm tạo ra chữ ký là hàm tính toán chữ ký trên cơ sở khóa mật và dữ liệu cần ký.
- Hàm kiểm tra chữ ký là hàm kiểm tra xem chữ ký đã cho có đúng với khóa công cộng không. Khóa này mọi người có quyền truy cập cho nên mọi người đều có thể kiểm tra được chữ ký.

Định nghĩa: Sơ đồ chữ ký bao gồm các thành phần sau:

- Không gian bản rõ M.
- Không gian chữ ký S.
- Không gian khóa K để tạo nên chữ ký, không gian khóa K’ để kiểm tra chữ ký.
- Thuật toán hiệu quả để tạo nên khóa Gen: , ở đây K và K’ tương ứng với không gian khóa mật và khóa công cộng.
- Thuật toán tạo chữ ký Sign: .
- Thuật toán kiểm tra chữ ký Verify: .

Đối với bất kỳ khóa tạo chữ ký và bất kỳ bản tin lệnh ký bức điện được ký hiệu:

$$s \leftarrow Sign_{sk}(m)$$

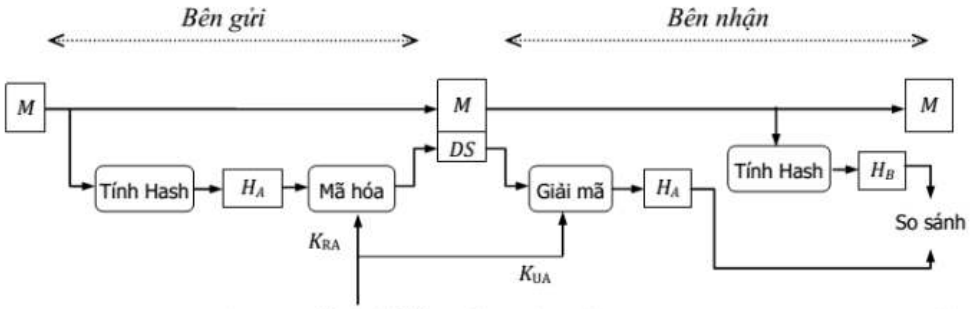
Biểu thức này được đọc như sau: s-là chữ ký của bản tin m được tạo ra nhờ thuật toán **Sign** và khóa mật **sk**.

Đối với bất kỳ khóa mật của chữ ký $sk \in K$, tương ứng với khóa công cộng để kiểm tra chữ ký là $pk \in K'$, bất kỳ bản tin $m \in M$ và chữ ký $s \in S$ cần thỏa mãn điều kiện sau:

$$Verify_{pk(m,s)} = \begin{cases} True, if & s = Sign_{sk(m)} \\ False, if & s \neq Sign_{sk(m)} \end{cases}$$

Bởi vì tài liệu cần ký thường có chiều dài khá dài. Một biện pháp để ký là chia tài liệu ra các đoạn nhỏ và sau đó ký lên từng đoạn và ghép lại. Nhưng phương pháp có nhược điểm là chữ ký lớn, thứ hai là ký chậm vì hàm ký là các hàm mũ, thứ ba là chữ ký có thể bị đảo loạn các vị trí không đảm tính nguyên vẹn của tài liệu. Chính vì điều đó mà khi ký thì người ta ký lên giá trị hàm hash của tài liệu, vì giá trị của hàm hash luôn cho chiều dài xác định. Hàm hash sẽ được xem trong chương sau.

Có nhiều cách để tạo ra chữ kí. Ta có thể sử dụng một sơ đồ sau:



Hình 1: Kiến trúc chữ ký điện tử tổng quát

1.3.4 Ưu điểm của chữ ký số

Việc sử dụng chữ ký số mang lại nhiều ưu điểm khi cần xác định nguồn gốc và tính toàn vẹn của văn bản trong quá trình sử dụng.

- **Khả năng xác định nguồn gốc**

  - Các hệ thống mật mã hóa khóa công khai cho phép mật mã hóa văn bản với khóa bí mật mà chỉ có người chủ của khóa biết.
  - Để sử dụng Chữ ký số thì văn bản cần phải được mã hóa hàm băm (là giải thuật nhằm sinh ra các giá trị băm tương ứng với mỗi khối dữ liệu: có thể là một chuỗi kí tự, một đối tượng trong lập trình hướng đối tượng, v.v.... Giá trị băm đóng vai gần như một khóa để phân biệt các khối dữ liệu). Sau đó dùng khoá bí mật của người chủ khóa để mã hóa, khi đó ta được Chữ ký số. Khi cần kiểm tra, bên nhận giải mã với khóa công khai để lấy lại hàm băm và kiểm tra với hàm băm của văn bản nhận được. Nếu hai giá trị này khớp nhau thì bên nhận có thể tin tưởng rằng văn bản đó xuất phát từ người sở hữu khóa bí mật.

- **Tính toàn vẹn**
  - Cả hai bên tham gia vào quá trình thông tin đều có thể tin tưởng là văn bản không bị sửa đổi trong khi truyền vì nếu văn bản bị thay đổi thì hàm băm cũng sẽ thay đổi và lập thức bị phát hiện. Quy trình mã hóa sẽ ẩn nội dung dối với bên thứ ba.
- **Tính không thể phủ nhận**
  - Trong giao dịch, một bên có thể từ chối nhận một văn bản nào đó là do mình gửi. Để ngăn ngừa khả năng này, bên nhận có thể yêu cầu bên gửi phải gửi kèm chữ ký số với văn bản. Khi có tranh chấp, bên nhận sẽ dùng chữ ký này như một chúng cứ để bên thứ ba giải quyết.

1.3.5 Sử dụng chữ ký số

- **Tạo chữ ký số**

Sử dụng các ứng dụng hỗ trợ tạo chữ ký số từ khóa bí mật, khóa bí mật do nhà cung cấp dịch vụ chứng thức chữ ký số công cộng cấp được lưu giữ dưới dạng tệp tin (có mật khẩu khi sử dụng), để an toàn và chống copy khóa bí mật một số nhà cung cấp dịch vụ lưu trữ khóa bí mật trong một thiết bị phần cứng chuyên dụng là USB Token hoặc SmartCard. Thiết bị này sẽ đảm bảo khóa bí mật được lưu trữ an toàn, không thể sao chép hay nhân bản được và cũng không thể bị virus phá hỏng.

- **Kiểm tra chữ ký**

Khi giao dịch điện tử, người nhận phải kiểm tra được tính pháp lý của chữ ký số của người giao dịch với mình gửi đến. Trong các ứng dụng hỗ trợ ký số có chức năng kiểm tra được chữ ký số công cộng hợp pháp hay không. Việc kiểm tra là so sánh tính đồng nhất của khóa công khai trên chữ ký số của người gửi đến với khóa công khai của Nhà cung cấp dịch vụ chứng thực chữ ký số công cộng lưu trữ trên hệ thống máy chủ của Trung tâm chứng thực chữ ký số quốc gia (Root Certification Authority) thuộc Bộ Thông tin – Truyền thông.

Chương 2. Kết quả nghiên cứu

Nhiệm vụ đề tài:

Tìm hiểu các kiến thức liên quan đến chữ ký RSA và xây dựng các chương trình ứng dụng tương ứng với các ngôn ngữ đã chọn trong phiếu phân công bài tập lớn.

Công việc chính:

1. Tìm hiểu về hệ mật mã RSA
2. Tìm hiểu về chữ ký điện tử sử dụng hệ mã RSA
3. Nghiên cứu về hàm băm mật mã và hàm băm MD5
4. Tìm hiểu về các ngôn ngữ lập trình C#, C++, Java và Python
5. Xây dựng chương trình Demo

2.1 Giới thiệu

2.1.1 Tên đề tài thực hiện

Tìm hiểu về chữ ký điện tử RSA và viết ứng dụng minh họa

2.1.2 Các bước thực hiện triển khai đề tài

a. Hình thức sản phẩm

Ứng dụng được viết bằng các ngôn ngữ lập trình đã được học trong các học phần chuyên ngành trước: Java, C++, Python, C#.

b. Kết quả đạt được

- Nghiên cứu tài liệu ATBMTT và hiểu được vấn đề đề tài đặt ra
- Nắm được các phương pháp mã hóa RSA và các vấn đề liên quan
- Hiểu được các giai đoạn thực hiện mã hóa theo hệ mã RSA
- Nắm được thuật toán và cách sử dụng các ngôn ngữ Java, Python, C++, C#

2.2 Nội dung thuật toán

<u>Mô tả thuật toán</u>

Bài toán sử dụng hệ mã đã học để tạo chữ ký điện tử RSA trải qua các bước cụ thể:

- Tạo khóa
  Xác định khóa công khai $K_{pub} = \{ b, n \}$
  Xác định khóa bí mật $K_{pr} = \{ a, p, q \}$
- Mã hóa: Sử dụng khóa K_{pub}
  $y = e_{kpub}(x) = x^b \bmod n$
- Giải mã: Sử dụng khóa K_{pr}
  $x = d_{kpr}(y) = y^a \bmod n$

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

Chương trình Java

Chương trình C++

Chương trình C#

Chương trình Javascript

Chương trình Python

2.4 Cài đặt và triển khai

Giới thiệu công cụ:

- **Visual Studio Code**
  - Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.
  - Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan

cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

- Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C,[4] C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010[5]). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.
- Microsoft cung cấp phiên bản "Express" (đối với phiên bản Visual Studio 2013 trở về trước) và "Community" (đối với bản Visual Studio 2015 trở về sau) là phiên bản miễn phí của Visual Studio.

- **Eclipse IDE**
  - Eclipse là 1 công cụ hỗ trợ lập trình mã nguồn mở được phát triển bởi IBM.
  - Eclipse IDE là một môi trường phát triển tích hợp (IDE) cho Java và các ngôn ngữ lập trình khác như C , C ++, PHP, và Ruby ... Môi trường phát triển được cung cấp bởi Eclipse bao gồm các công cụ phát triển Java Eclipse (JDT) cho Java, Eclipse CDT cho C/C ++, và Eclipse PDT cho PHP, và một số thứ khác.
  - Nền tảng Eclipse và các plugin khác từ nền tảng Eclipse được phát hành theo giấy phép **Eclipse Public License (EPL)**. EPL đảm bảo rằng Eclipse được tải xuống và cài đặt **hoàn toàn miễn phí**. Nó cũng cho phép Eclipse được sửa đổi và phân phối bởi cộng đồng.
    - Tính năng chính của Eclipse IDE
    - Hỗ trợ nhiều loại ngôn ngữ lập trình.
    - Chỉnh sửa mã nguồn thông minh.
    - Giao diện trực quan, dễ thao tác, sử dụng,
    - Là một công cụ lập trình phần mềm máy tính hoặc phần mềm trên các thiết bị di động.
    - Gỡ lỗi mạng nội bộ và từ xa.
    - Thử nghiệm tính năng xây dựng giao diện đồ họa.
    - Tính năng Quick Search (Tìm kiếm nhanh), tự động biên dịch, hỗ trợ các Framework cho website, trình ứng dụng máy chủ GlassFish và cơ sở dữ liệu.

2.5 Thực hiện bài toán

2.5.1 Phân công công việc

Bảng phân công công việc của nhóm:

Tên sinh viên	Tên công việc
Nguyễn Đức Đạt	- Giới thiệu, đặc điểm, tính chất về hàm băm - Hàm băm MD5 và ứng dụng - Thuật toán MD5 - Viết ứng dụng bằng ngôn ngữ C++
Nguyễn Hữu Đạt	- Tìm hiểu về chữ ký điện tử - Chữ ký điện tử RSA - Viết ứng dụng bằng ngôn ngữ JavaScript
Nguyễn Tuấn Đạt	- Các loại tấn công - Viết ứng dụng bằng ngôn ngữ Java
Nguyễn Thế Đoàn	- Mã hóa bất đối xứng - Lợi ích và hạn chế của mã hóa bất đối xứng - Viết chương trình bằng ngôn ngữ Python
Phạm Đăng Đông	- Ứng dụng của mã hóa bất đối xứng - Viết chương trình bằng ngôn ngữ C#

2.5.2 Lê Văn Hà – Các nội dung tìm hiểu

2.5.2.1 Tìm hiểu về hàm băm MD5

a. Giới thiệu

- Hàm băm là các thuật toán không sử dụng khóa để mã hóa, nó có nhiệm vụ "lọc" (băm) thông điệp được đưa vào vào theo một thuật toán h một chiều nào đó, rồi đưa ra một bản băm – văn bản đại diện – có kích thước cố định. Do đó người nhận không biết được nội dung hay độ dài ban đầu của thông điệp đã được băm bằng hàm băm.
- Giá trị của hàm băm là duy nhất, và không thể suy ngược lại được nội dung thông điệp từ giá trị băm này.
- Việc sử dụng các hệ mật mã và các sơ đồ chữ ký số, thường là mã hóa và ký số trên từng bit của thông tin, sẽ tỷ lệ với thời gian để mã hóa và dung lượng của thông tin.
- Thêm vào đó có thể xảy ra trường hợp: Với nhiều bức thông điệp đầu vào khác nhau, sử dụng hệ mật mã, sơ đồ ký số giống nhau (có thể khác nhau) thì cho ra kết quả bản mã, bản ký số giống nhau (ánh xạ N-1: nhiều – một). Điều này sẽ dẫn đến một số rắc rối về sau cho việc xác thực thông tin.

⇨ Vì vậy, giarp pháp cho các vấn đề vướng mắc đến chữ ký số là dùng hàm băm để trợ giúp cho việc ký số

Các thuật toán băm với đầu vào là các bức thông điệp có dung lượng, kích thước tùy ý (vài KB đến vài chục MB thậm chí hơn nữa) – các bức thông điệp có thể là dạng văn bản, hình ảnh, âm thanh, file ứng dụng v.v… - và với các thuật toán băm: MD2, MD4, MD5, SHA cho các bản băm đầu ra có kích thước cố định: 128 bit với dòng MD, 160 bit với SHA. Như vậy, bức thông điệp kích thước tùy ý sau khi băm sẽ được thu gọn thành những bản băm – được gọi là các "văn bản đại diện" – có kích thước cố định (128 bit hoặc 160 bit).

b. Đặc điểm

Với mỗi thông điệp đầu vào chỉ có thể tính ra được một văn bản đại diện – giá trị băm tương ứng – duy nhất.

Hai thông điệp khác nhau chắc chắn có hai văn bản đại diện khác nhau. Khi đã có văn bản đại diện duy nhất cho bức thông điệp, áp dụng các sơ đồ chữ ký số ký trên văn bản đại diện đó.

c. Tính chất của hàm băm

Tính chất 1: Hàm hash h là hàm không va chạm yếu nếu khi cho trước một bức điện x, không thể tiến hành về mặt tính toán để tìm một bức điện x x' sao cho h (x') = h(x).

Tính chất 2: Hàm Hash h là không va chạm mạnh nếu không có khả năng tính toán để tìm ra bức điênk x và x' sao cho x ≠ x' và h(x) = h(x').

Tính chất 3: Hàm Hash h là một chiều nếu khi cho trước một bản tóm lược thông báo z, không thể thực hiện về mặt tính toán để tìm bức điện x sao cho h(x) = z.

2.5.2.2 Hàm băm MD5 và ứng dụng

MD5 (Message-Digest algorithm 5) là một hàm băm để mã hóa với giá trị băm là 128bit. Từng được xem là một chuẩn trên Internet, MD5 đã được sử dụng rộng rãi trong các chương trình an ninh mạng, và cũng thường được dùng để kiểm tra tính nguyên vẹn của tập tin. Nó đã được sử dụng rộng rãi trong các chương trình an ninh mạng, và cũng thường được dùng để kiểm tra tính nguyên vẹn của tập tin. MD5 được thiết kế bởi Ronald Rivest vào năm 1991 để thay thế cho hàm băm trước đó MD4.

Có 2 ứng dụng quan trọng :

- MD5 được sử dụng rộng rãi trong thế giới phần mềm để đảm bảo rằng tập tin tải về không bị hỏng. Người sử dụng có thể so sánh giữa thông số kiểm tra phần mềm bằng MD5 được công bố với thông số kiểm tra phần mềm tải về bằng MD5. Hệ điều hành Unix sử dụng MD5 để kiểm tra các gói mà nó phân phối, trong khi hệ điều hành Windows sử dụng phần mềm của hãng thứ ba.
- MD5 được dùng để mã hóa mật khẩu. Mục đích của việc mã hóa này là biến đổi một chuỗi mật khẩu thành một đoạn mã khác, sao cho từ đoạn mã đó không thể nào lần trở lại mật khẩu. Có nghĩa là việc giải mã là không thể hoặc phải mất một khoảng thời gian vô tận (đủ để làm nản lòng các hacker).

2.5.2.3 Thuật toán MD5

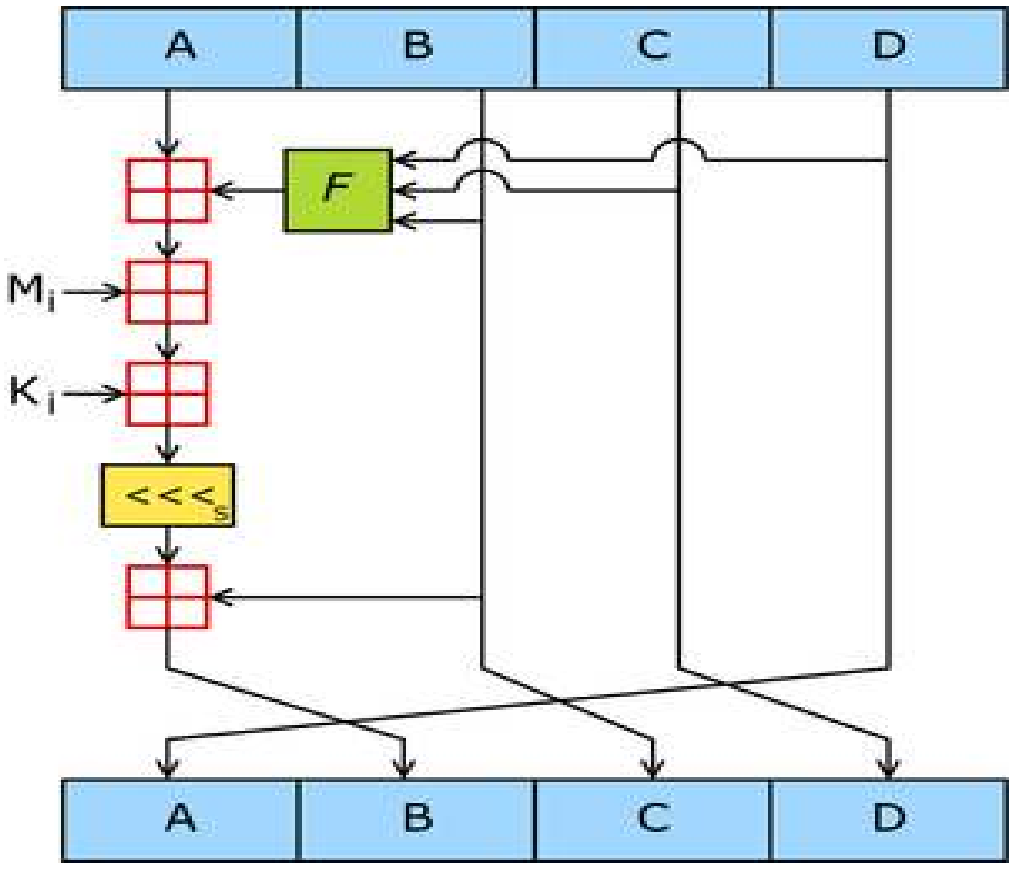
MD5 biến đổi một thông điệp có chiều dài bất kì thành một khối có kích thước cố định 128 bits. Thông điệp đưa vào sẻ được cắt thành các khối 512 bits. Thông điệp được đưa vào bộ đệm để chiều dài của nó sẻ chia hết cho 512.

Bộ đệm hoạt động như sau:

- Trước tiên nó sẽ chèn bit 1 vào cuối thông điệp
- Tiếp đó là hàng loạt bit Zero cho tới khi chiều dài của nó nhỏ hơn bội số của 512 một khoảng 64 bit.
- Phần còn lại sẻ được lấp đầy bởi một số nguyên 64 bit biểu diễn chiều dài ban đầu của thông điệp.

Thuật toán chính của MD5 hoạt động trên một bộ 128 bit. Chia nhỏ nó ra thành 4 từ 32 bit, kí hiệu là A,B,C và D. Các giá trị này là các hằng số cố định. Sau đó thuật toán chính sẻ luân phiên hoạt động trên các khối 512 bit. Mỗi khối sẽ phối hợp với một bộ. Quá trình xữ lý một khối thông điệp bao gồm 4 bước tương tự nhau, gọi là vòng ("round"). Mỗi vòng lại gồm 16 quá trình tương tự nhau dựa trên hàm một chiều F, phép cộng module và phép xoay trái…

Đây là hình mô tả một quá trình trong một vòng. Có 4 hàm một chiều F có thể sử dụng. Mỗi vòng sử dụng một hàm khác nhau.



Hình 2: Quá trình trong 1 vòng

Mặc dù MD4 vẫn chưa bị phá, song các phiên bản yếu cho phép bỏ qua hoặc vòng thứ nhất hay thứ ba đều có thể bị phá không khó khăn gì. Nghĩa là dễ dàng tìm thấy các va chạm đối với các phiên bản chỉ có hai vòng. Phiên bản mạnh của dòng MD là MD5 được công bố năm 1991. MD5 dùng bốn vòng thay cho ba và chậm hơn 30% so với MD4 (khoảng 0,9 MB/giây trên cùng một máy).

Mô tả thuật toán

Input : Thông điệp (văn bản) có độ dài tùy ý.

Output : Bản băm, đại diện cho thông điệp gốc, đồ dài cố định 128 bit.

Giả sử đầu vào là một xâu ***a*** có độ dài ***b*** bit (***b*** có thể bằng 0)

Bước 1*:* Khởi tạo các thanh ghi

Có 4 thanh ghi được sử dụng để tính toán nhằm đưa ra các đoạn mã: A, B, C, D. Bản tóm lược của thông điệp được xây dựng như sự kết nối của các thanh ghi. Mỗi thanh ghi có độ dài 32 bit. Các thanh ghi này được khởi tạo giá trị hecxa.

word A := 67 45 23 01
word B := EF CD AB 89
word C := 98 BA DC FE
word D := 10 32 54 76

Bước 2+3*:*

Xử lý thông điệp ***a*** trong 16 khối *word*, có nghĩa là xử lý cùng một lúc 16 *word* = 512 bit (chia mảng M thành các khối 512 bit, đưa từng khối 512 bit đó vào mảng T[j]). Mỗi lần xử lý một khối 512 bit. Lặp lại N/16 lần.

Bước 4*:* Thực hiện bốn vòng băm

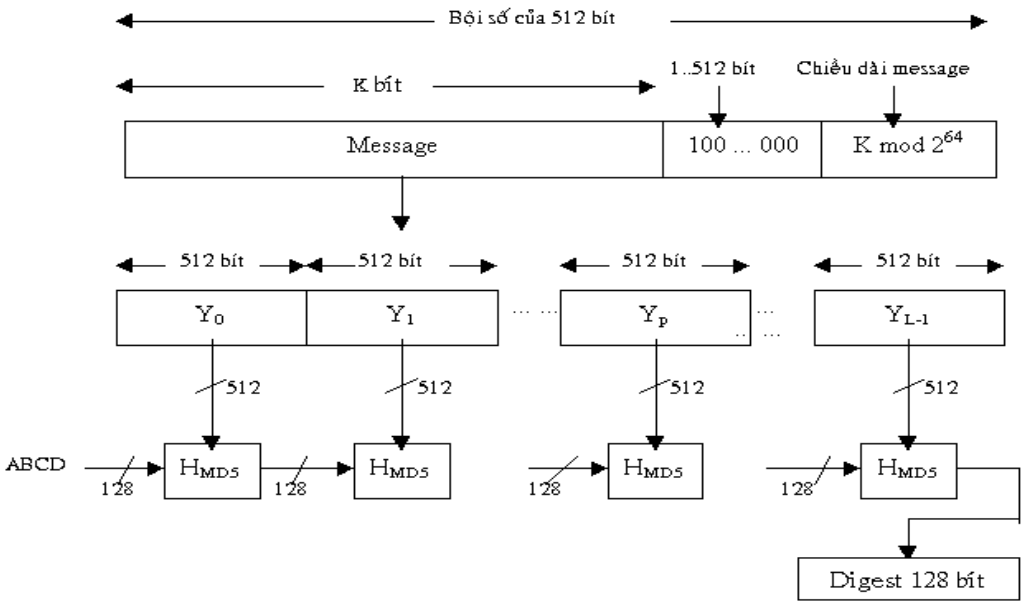
Các vòng 1, 2, 3 và 4 dùng tương ứng ba hàm F, G, H và I. Mỗi hàm này là một hàm boolean tính theo bit. Chúng được xác định như sau:

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

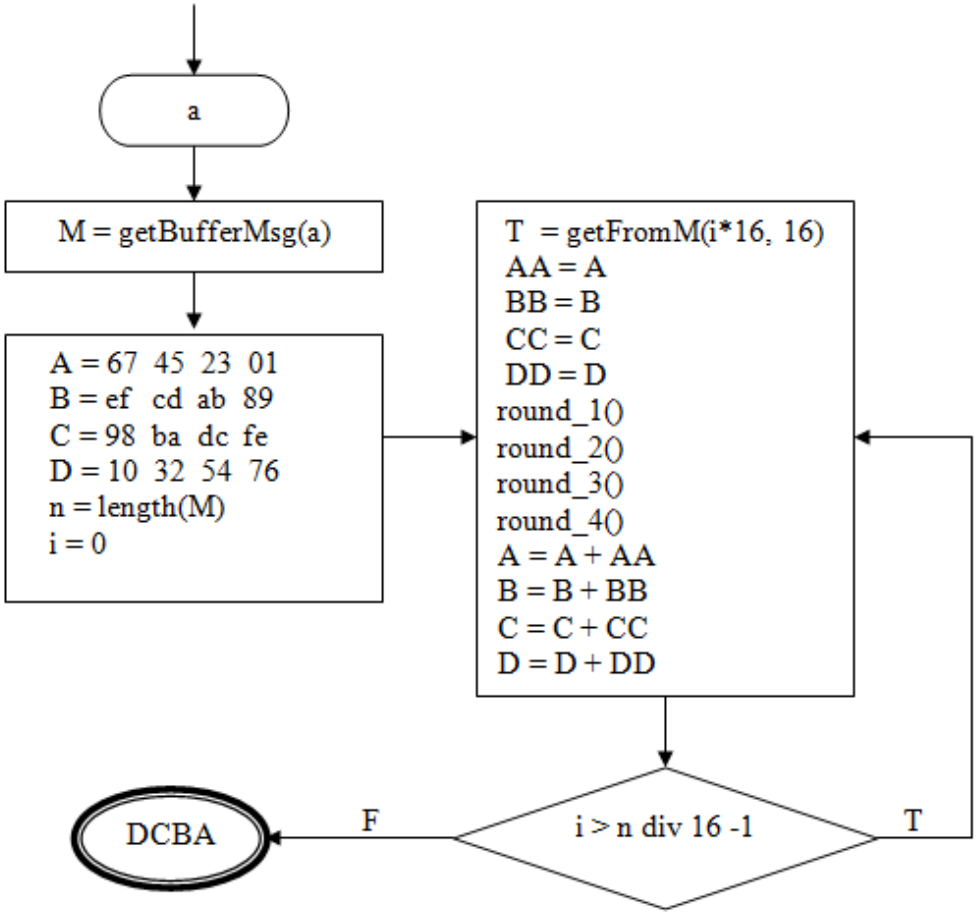
$$I(X, Y, Z) = Y \oplus (X \vee (\neg Z))$$



Hình 3: Quá trình tạo bản băm của MD5

Thuật toán MD5

```
1.    A := 67  45  23  01
      B := ef  cd  ab  89
      C := 98  ba  dc  fe
      D := 10  32  54  76
2.    for  i := 0  to  n/16 - 1  do {
3.          for  j := 0  to  15  do  T[j] = M[16i + j];
4.    AA := A;    Mỗi lần xử lý 16 từ, mỗi từ 32 bit,  tl: 512  bit.
                        BB := B;
      CC := C;
      DD := D;
5.          round_1();
6.          round_2();
7.          round_3();
8.          round_4();
9.          A = A + AA
            B = B + BB
            C = C + CC
            D = D + DD
}
```



Hình 4: Lược đồ thuật toán MD5

Bốn vòng trong MD5 là hoàn toàn khác nhau. Mỗi vòng (5, 6, 7, 8) gồm một trong 16 *word* trong T.

MD5 sử dụng thêm một mảng S[1 … 64] được xây dựng từ hàm *sin*. Với S[i], giá trị của phần tử thứ i trong mảng S, tương đương với phần nguyên của 4294967296 × abs(sin(i)), với i tính theo radian.

- Vòng 1 sử dụng giá trị trong mảng S[1 … 16]
- Vòng 2 sử dụng giá trị trong mảng S[17 … 32]
- Vòng 3 sử dụng giá trị trong mảng S[33 … 48]
- Vòng 4 sử dụng giá trị trong mảng S[49 … 64]

Ta có mảng S[1 … 64] như sau (tất cả đều để ở hệ cơ số 16):

1. D76AA478	17. F61E2562	33. FFFA3942	49. F4292244
2. E8C7B756	18. D040B340	34. 8771F681	50. 432AFF97
3. 242070DB	19. 265E5A51	35. 6D9D6122	51. AB9423A7

4. C1BDCEEE	20. E9B6C7AA	36. FDE5390C	52. FC93A039
5. F57C0FAF	21. D62F105D	37. A4BEEA44	53. 655B59C3
6. 4787C62A	22. 02441453	38. 4BDECFA9	54. 8F0CCC92
7. A8304613	23. D8A1E681	39. F6BB4B60	55. FFEFF47D
8. FD469501	24. E7D3FBC	40. BEBFBC70	56. 85845DD1
9. 698098D8	25. 21E1CDE6	41. 289B7EC6	57. 6FA87E4F
10. 8B44F7AF	26. C33707D6	42. EAA127FA	58. FE2CE6E0
11. FFFF5BB1	27. F4D50D87	43. D4EF3085	59. A3014314
12. 895CD7BE	28. 455A14ED	44. 04881D05	60. 4E0811A1
13. 6B901122	29. A9E3E905	45. D9D4D039	61. F7537E82
14. FD987193	30. FCEFA3F8	46. E6DB99E5	62. BD3AF235
15. A679438E	31. 676F02D9	47. 1FA27CF8	63. 2AD7D2BB
16. 49B40821	32. 8D2A4C8A	48. D4AC5665	64. EB86D391

Giá trị hằng số của mảng S có 64 phần tử

Các phép toán được thực hiện trong bốn vòng tạo ra các giá trị mới trong bốn *thanh ghi*. Cuối cùng, bốn *thanh ghi* được cập nhật ở 3.5 bằng cách cộng ngược các giá trị lưu trước đó ở 2.3. Phép cộng này được xác định là cộng các số nguyên dương, được rút gọn theo modulo 2^{32}.

Vòng 1 (*round_1()*)	Vòng 2 (*round_2()*)
1. A = (A + F(B, C, D) + T0] + S[1] <<< 7	1. A = (A + G(B, C, D) + T[1] + S[17] <<< 5
2. D = (D + F(A, B, C) + T[1] + S[2]) <<< 12	2. D = (D + G(A, B, C) + T[6] + S[18]) <<< 9
3. C = (C + F(D, A, B) + T[2] + S[3]) <<< 17	3. C = (C + G(D, A, B) + T[21] + S[19]) <<< 14
4. B = (B + F(C, D, A) + T[3] + S[4]) <<< 22	4. B = (B + G(C, D, A) + T[0] + S[20]) <<< 20
5. A = (A + F(B, C, D) + T[4] + S[5]) <<< 7	5. A = (A + G(B, C, D) + T[5] + S[21]) <<< 5
6. D = (D + F(A, B, C) + T[5] + S[6]) <<< 12	6. D = (D + G(A, B, C) + T[10] + S[22]) <<< 9
7. C = (C + F(D, A, B) + T[6] + S[7]) <<< 17	7. C = (C + G(D, A, B) + T[15] + S[23]) <<< 14
8. B = (B + F(C, D, A) + T[7] + S[8]) <<< 22	8. B = (B + G(C, D, A) + T[4] + S[24]) <<< 20
9. A = (A + F(B, C, D) + T[8] + S[9]) <<< 7	9. A = (A + G(B, C, D) + T[9] + S[25]) <<< 5
10. D = (D + F(A, B, C) + T[9] + S[10]) <<< 12	10.D = (D + G(A, B, C) + T[14] + S[26]) <<< 9
11. C = (C + F(D, A, B) + T[10] + S[11]) <<< 17	11.C = (C + G(D, A, B) + T[3] + S[27]) <<< 14
12. B = (B + F(C, D, A) + T[11] + S[12]) <<< 22	12.B = (B + G(C, D, A) + T[8] + S[28]) <<< 20
13. A = (A + F(B, C, D) + T[12] + S[13]) <<< 7	13.A = (A + G(B, C, D) + T[13] + S[29]) <<< 5
14. D = (D + F(A, B, C) + T[13] + S[14]) <<< 12	14.D = (D + G(A, B, C) + T[2] + S[30]) <<< 9
15. C = (C + F(D, A, B) + T[14] + S[15]) <<< 17	15.C = (C + G(D, A, B) + T[7] + S[31]) <<< 14
16. B = (B + F(C, D, A) + T[15] + S[16]) <<< 22	16.B = (B + G(C, D, A) + T[12] + S[32]) <<< 20

Vòng 3 (*round_1()*)	Vòng 4 (*round_4()*)
1. A = (A + H(B, C, D) + T[5] + S[33]) <<< 4	1. A = (A + I(B, C, D) + T[0] + S[49]) <<< 6
2. D = (D + H(A, B, C) + T[8] + S[34]) <<< 11	2. D = (D + I(A, B, C) + T[7] + S[50]) <<< 10
3. C = (C + H(D, A, B) + T[11] + S[35]) <<< 16	3. C = (C + I(D, A, B) + T[14] + S[51]) <<< 15
4. B = (B + H(C, D, A) + T[14] + S[36]) <<< 23	4. B = (B + I(C, D, A) + T[5] + S[52]) <<< 21
5. A = (A + H(B, C, D) + T[1] + S[37]) <<< 4	5. A = (A + I(B, C, D) + T[12] + S[53]) <<< 6
6. D = (D + H(A, B, C) + T[4] + S[38]) <<< 11	6. D = (D + I(A, B, C) + T[3] + S[54]) <<< 10
7. C = (C + H(D, A, B) + T[7] + S[39]) <<< 16	7. C = (C + I(D, A, B) + T[10] + S[55]) <<< 15
8. B = (B + H(C, D, A) + T[10] + S[40]) <<< 23	8. B = (B + I(C, D, A) + T[1] + S[56]) <<< 21
9. A = (A + H(B, C, D) + T[13] + S[41]) <<< 4	9. A = (A + I(B, C, D) + T[8] + S[57]) <<< 6
10. D = (D + H(A, B, C) + T[0] + S[42]) <<< 11	10. D = (D + I(A, B, C) + T[15] + S[58]) <<< 10
11. C = (C + H(D, A, B) + T[3] + S[43]) <<<16	11. C = (C + I(D, A, B) + T[6] + S[59]) <<< 15
12. B = (B + H(C, D, A) + T[6] + S[44]) <<< 23	12. B = (B + I(C, D, A) + T[13] + S[60]) <<< 21
13. A = (A + H(B, C, D) + T[9] + S[45]) <<< 4	13. A = (A + I(B, C, D) + T[4] + S[61]) <<< 6
14. D = (D + H(A, B, C) + T[12] + S[46]) <<< 11	14. D = (D + I(A, B, C) + T[11] + S[62]) <<< 10
15. C = (C + H(D, A, B) + T[15] + S[47]) <<< 16	15. C = (C + I(D, A, B) + T[2] + S[63]) <<< 15
16. B = (B + H(C, D, A) + T[2] + S[48]) <<< 23	16. B = (B + I(C, D, A) + T[9] + S[64]) <<< 21

Bước 5: Output

Kết quả ra là đoạn mã có độ dài 128 bit, được thu gọn từ thông điệp ***a*** có

độ dài ***b*** bit. Đoạn mã này thu được từ 4 thanh ghi A, B, C, D: bắt đầu từ

byte thấp của thanh ghi A cho đến byte cao của thanh ghi D.

Hàm băm MD5 (còn được gọi là hàm tóm tắt thông điệp - message degests) sẻ trả về một chuỗi số thập lục phân gồm 32 số liên tiếp. Dưới đây là các ví dụ mô tả các kết quả thu được sau khi băm.

MD5("cộng hòa xã hội chủ nghĩa việt nam")

= 7b8e76fac176d53c53cb24843e31e759

Thậm chí chỉ cần một thay đổi nhỏ cũng làm thay đổi hoàn toàn kết quả trả về :

MD5(“ Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam “)

= 0634f131b89616154a643be79b61eda4

Ngay cả một chuỗi rỗng cũng cho ra một kết quả phức tạp:
MD5(“”)

= d41d8cd98f00b204e9800998ecf8427e

Ưu điểm so với MD4:

1. Một vòng thứ tư đã được thêm vào.

2. Mỗi bước của chu kì, MD5 sử dụng một hằng số phân biệt, trong khi MD4 sử dụng hằng số chung cho mọi thao tác trong cùng chu kì không đổi.

3. Các chức năng ở vòng 2 đã được thay đổi từ (XY v XZ v YZ) để (XZ v Y not (Z)) để làm giảm tính đối xứng.

4. Mỗi bước bây giờ có thêm trong kết quả của bước trước. Điều này thúc đẩy nhanh hơn hiệu ứng lan truyền Avalanche.

5. Thứ tự từ đầu vào được truy cập trong vòng 2 và 3 là thay đổi, để làm cho các mô hình nhỏ như nhau.

6. Các hệ số dịch chuyển xoay vòng trong mỗi chu kỳ được tối ưu hóa nhằm tăng tốc độ hiệu ứng lan truyền. Ngoài ra, mỗi chu kỳ sử dụng 4 hệ số dịch chuyển khác nhau.

2.5.2.4 Viết ứng dụng bằng ngôn ngữ Java

a. Ngôn ngữ Java là gì?

Java là một ngôn ngữ lập lập trình, được phát triển bởi Sun Microsystem vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

b. Ứng dụng của ngôn ngữ Java

- Phát triển ứng dụng cho các thiết bị điện tử thông minh, các ứng dụng cho doanh nghiệp với quy mô lớn.
- Tạo các trang web có nội dung động (web applet), nâng cao chức năng của server.
- Phát triển nhiều loại ứng dụng khác nhau: Cơ sở dữ liệu, mạng, Internet, viễn thông, giải trí,...

c. Những đặc điểm cơ bản của Java

- Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là "Write Once, Run Anywhere" (Viết một lần, chạy mọi nơi), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, chúng ta viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS...

- Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:
    - Đơn giản và quen thuộc: Vì Java kế thừa trực tiếp từ C/C++ nên nó có những đặc điểm của ngôn ngữ này, Java đơn giản vì mặc dù dựa trên cơ sở C++ nhưng Sun đã cẩn thận lược bỏ các tính năng khó nhất của của C++ để làm cho ngôn ngữ này dễ sử dụng hơn.
    - Hướng đối tượng và quen thuộc.
    - Mạnh mẽ (thể hiện ở cơ chế tự động thu gom rác - Garbage Collection) và an toàn.
    - Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (Portability).
    - Hiệu suất cao.
    - Máy ảo (biên dịch và thông dịch).
    - Phân tán.
    - Đa nhiệm: Ngôn ngữ Java cho phép xây dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.

d. Các platform cơ bản của Java

Java Platform gồm có 3 thành phần chính:

- Java Virtual Machine (Java VM): Máy ảo Java
- Java Application Programming Interface (Java API).
- Java Development Kit (JDK) gồm trình biên dịch, thông dịch, trợ giúp, soạn tài liệu... và các thư viện chuẩn

e. Tiêu chuẩn một môi trường Java điển hình

Thông thường, các chương trình Java trải qua 5 giai đoạn chính:

- Editor: Lập trình viên viết chương trình và được lưu vào máy tính với định dạng .java.
- Compiler: Biên dịch chương trình thành bytecodes (định dạng .class) - nhờ bước trung gian này mà Java được viết 1 lần và chạy trên các hệ điều hành khác nhau.
- Class Loader: Đọc file .class chứa mã bytecodes và lưu vào trong bộ nhớ.
- Bytecode Verifier: Đảm bảo rằng mã bytecodes là hợp lệ và không vi phạm các vấn đề về bảo mật của Java.

- Intepreter: Biên dịch bytecodes thành mã máy để máy tính có thể hiểu được và sau đó thực thi chương trình.

f. Chương trình ứng dụng ngôn ngữ Java

File chương trình đính kèm trong báo cáo bài tập lớn

2.5.3 Lê Minh Hiển – Các nội dung tìm hiểu

2.5.3.1 Tìm hiểu về chữ ký điện tử, chữ ký điện tử RSA

- Chữ kí điện tử (Digital Signature) là một chuỗi dữ liệu liên kết với một thông điệp (message) và thực thể tạo ra thông điệp.
- Giải thuật tạo ra chữ ký điện tử (Digital Signature generation algorithm) là một phương pháp sinh chữ ký điện tử.
- Giải thuật kiểm tra chữ ký điện tử (Digital Signature verification algorithm) là một phương pháp xác minh tính xác thực của chữ ký , có nghĩa là nó thực sự được tạo ra bởi 1 bên chỉ định.
- Một hệ chữ ký điện tử (Figital Signature Scheme) bao gồm giải thuật tạo chữ số và giải thuật kiểm tra chữ kỹ số.
- Quá trình tạo chữ ký điện tử (Digital Signature signing process) bao gồm:
  - Giải thuật tạo chữ ký điện tử.
  - Phương pháp chuyển dữ liệu thông điệp thành dạng có thể ký được
- Quá trình kiểm tra chữ ký điện tử (Digital signature verification process) bao gồm:
  - Giải thuật kiểm tra chữ ký điện tử.
  - Phương pháp khôi phục dữ liệu từ thông điệp

 RSA :

- RSA là thuật toán mã hoá khoá công khai.
- Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hoá.
- RSA đang được sử dụng phổ biến trong thương mại điện tử được đánh giá là an toàn với điều kiện độ dài khoá đủ lớn.
- RSA được xây dựng dựa trên độ khó của bài toán phân tích các số lớn ra thừa số nguyên tố: Biết một số nguyên tố nhân chúng với nhau để thu được một hợp số là dễ

còn biết hợp số rồi phân tích nó ra thừa số nguyên tố khó.

- Lược đồ chữ ký điện tử RSA.
  - Trong phần này mô tả lược đồ chữ ký RSA. Độ an toàn của lược đồ chữ ký RSA dựa vào độ an toàn của hệ mã RSA. Lược đồ bao gồm cả chữ ký số kèm theo bản rõ và tự khôi phục thông điệp từ chữ ký số.
    - Thuật toán sinh khóa cho lược đồ chữ ký RSA
    - Thuật toán sinh chữ ký RSA
    - Thuật toán chứng thực chữ ký RSA

2.5.3.2 Viết ứng dụng bằng ngôn ngữ C++

a. Ngôn ngữ C++ là gì?

Ngôn ngữ lập trình C++ là một ngôn ngữ lập trình hướng đối tượng(OOP – Object-oriented programming) được phát triển bởi Bjarne Stroustrup. C++ là ngôn ngữ lập trình được phát triển trên nên tảng của ngôn ngữ lập trình C

b. Những phiên bản C++

Ngôn ngữ lập trình này được ISO công nhận chuẩn hóa đầu tiền vào năm 1998 với tên gọi là dạng ISO/ IEC 14882: 1998.
Sau đó là C++ 03, C++ 11 và C++ 14.
Tháng 12 năm 2017 tổ chức tiêu chuẩn quốc tế (ISO) đã công nhận và chuẩn hóa phiên bản mới nhất của C++ là ISO/ IEC 14882: 2017 (gọi tắt là C++ 17). Và kế hoạch tiếp theo sẽ là C++ 20.

c. Ứng dụng của ngôn ngữ C++

Ngôn ngữ C++ là phục vụ cho học lập trình cơ bản. Bởi vì đây là một ngôn ngữ lập trình bậc trung. Hầu hết các trường đào tạo công nghệ thông tin ở Việt Nam đều dùng 2 ngôn ngữ này làm môn cơ sở ngành

d. Chương trình ứng dụng ngôn ngữ C++

File chương trình đính kèm trong báo cáo bài tập lớn

2.5.4 Nguyễn Quỳnh Giao – Các nội dung tìm hiểu

2.5.4.1 Các dạng tấn công

a) Tấn công lặp

Simons và Norris đã chỉ ra rằng hệ thống RSA có thể bị tấn côn gkhi sử dụng tấn công lặp liên tiếp. Đó là khi kẻ tấn công biết khóa công khai (e, n) và bản mã C thì anh ta có thể tính chuỗi các bản mã sau:

$C1 = C^e \pmod{n}$ $C2 = C1^e \pmod{n}$ ……

$Ci = Ci\text{-}1^e \pmod{n}$

1.

Nếu có một phần tử Cj trong chuỗi C1, C2, …, C, … sao cho Cj = C thì khi đó anh ta sẽ tìm được M = Cj-1 bởi vì:

$Cj = C_{j\text{-}1} \pmod{n}$

$C = M^e \pmod{n}$

b) Kiểu tấn công module n dùng chung

Simons và Norris cũng chỉ ra rằng hệ thống RSA có thể bị tấn công khi sử dụng module n dùng chung, thực vậy nếu một thông điệp M được mã hóa bằng hai khóa công khai e1 và e từ hai thành viên trê hệ thống thì được:

$C1 = M^{e1} \pmod{n}$

$C2 = M^{e2} \pmod{n}$

Sau đó người tấn công dùng thuật toán Euclide mở rộng:

e1*a +e2*b = 1 sao cho gcd(e1, e2) = 1 thì M được khôi phục lại như sau: $M = C1^a \cdot C2^b \bmod n$.

c) Tấn công khi khóa công khai e nhỏ

- Hastad đã đưa ra kiểu tấn công khi khóa côn gkhai e nhỏ (e = 3) của hệ mã công khai RSA như sau:
  - Giả sử để gửi thông điệp M đến các người dùng P1, P2, …., Pk với khóa công khai là (e1, ni). A mã hóa M bằng khóa công khai (ei, ni) và gửi các bản mã Ci đến người dùng Pi, biết M<n với I = 1, 2,…, n
  - Ta có thể nghe trộm kết nối ra ngoài của A và thu thập được k bản mã Ci.

- Giả sử các khóa công khái ei = 3 thì có thể khôi phục M nếu k >= 3
- Thực vậy, nếu có được C1, C2, C3 với C1 = M^3 mod ni; C2= M^3 mod n2; C3= M^3 mod n3 và gcd(ni, nj) = 1, i ≠ j. Áp dụng định lý số dư Trung Hoa với C1,C2,C3 tìm được:C' thuộc Z_{n1n2n3} thỏa C' = M^3 mod $n_1n_2n_3$ -> M^3 là số nguyên

Vậy M=

2.5.4.2 Viết ứng dụng bằng ngôn ngữ Python

a. Ngôn ngữ Python là gì ?

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng. Ngôn ngữ lập trình Python được tạo bởi Guido van Rossum và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

b. Tính năng chính cảu Python

- ***Ngôn ngữ lập trình đơn giản, dễ học***: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

- ***Miễn phí, mã nguồn mở:*** Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

- ***Khả năng di chuyển***: Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

- ***Khả năng mở rộng và có thể nhúng***: Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

- ***Ngôn ngữ thông dịch cấp cao***: Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

- ***Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến***: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

- ***Hướng đối tượng:*** Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

c. Chương trình ứng dụng ngôn ngữ Python

File chương trình đính kèm trong báo cao bài tập lớn

2.5.5 Phạm Văn Giang & Nguyễn Mạnh Duy – Các nội dung tìm hiểu

2.5.5.1 Mã hóa bất đối xứng

Kiểu mã hóa này còn có tên gọi khác là mã hóa khóa công khai. Nó sử dụng đến hai khóa (key) khác nhau. Một khóa gọi là khóa công khai (public key) và một khóa khác là khóa bí mật (private key). Dữ liệu được mã hóa bằng public

key. Tất cả mọi người đều có thể có được key này. Tuy nhiên để giải mã được dữ liệu, người nhận cần phải có private key.

2.5.5.2 Cách hoạt động của mã hóa bất đối xứng

- Người nhận sẽ tạo ra một gặp khóa (public key và private key), họ sẽ giữ lại private key và truyền cho bên gửi public key. Vì public key này là công khai nên có thể truyền tự do mà không cần bảo mật.
- Trước khi gửi tin nhắn, người gửi sẽ mã hóa dữ liệu bằng mã hóa bất đối xứng với những key nhận được từ người nhận
- Người nhận sẽ giải mã dữ liệu nhận được bằng thuật toán được sử dụng ở bên người gửi, với key giải mã là private key.
- Điểm yếu lớn nhất của kiểu mã hóa này là tốc độ mã hóa và giải mã rất chậm. Nếu dùng kiểu mã hóa bất đối xứng trong việc truyền dữ liệu thì sẽ rất tốn phí và mất thời gian.

2.5.5.3 Ứng dụng của mã hóa bất đối xứng

- Tại sao cần mã hóa bất đối xứng ?Có thể sử dụng mã hóa bất đối xứng cho các hệ thống mà trong đó nhiều người dùng có thể cần mã hóa và giải mã tệp hoặc bộ dữ liệu, đặc biệt trong trường hợp không có giới hạn về tốc độ và sức mạnh tính toán. Ví dụ, có thể sử dụng phương thức mã hóa này trong hệ thống email được mã hóa, trong đó khóa công khai được sử dụng để mã hóa các email và khóa cá nhân được sử dụng để giải mã chúng.
- Hầu như các ứng dụng dùng hàng ngày hiện nay như:Facebook, Gmail, Amazon… đều sử dụng giao thức HTTPs .Có thể hiểu là giao thức HTTPs an toàn hơn HTTP vì toàn bộ thông tin truyền đi giữa client và server được bảo vệ bởi mã hóa SSL/TSL. SSL/TSL này hoạt động dựa trên cả hai loại mã hóa đối xứng và bất đối xứng .Nhờ nó mà chúng ta có thể đảm bảo bí mật khi thực hiện những giao thức giao dịch có chứa thông tin trong suốt quá trình truyền nhận dữ liệu .Có thể nói,nếu không có khóa bí mật ,đặc biệt là khóa bất đối xứng thì không có thương mại điện tử

2.5.5.4 Lợi ích và hạn chế của mã háo bất đối xứng

- **Lợi ích**

- Lợi ích rõ ràng nhất của loại mã hóa này là tính bảo mật của nó vì khóa riêng tư không cần phải được chuyển cho bất kỳ ai. Tất nhiên, điều này giúp đơn giản hóa đáng kể việc quản lý khóa trong các mạng lớn hơn.
- Về cơ bản,HTTP truyền dữ liệu dưới dạng plain text , nghĩa là có ai nghe lén dữ liệu bạn truyền và nhận với server thì có thể đọc và can thiệp được nội dung .Ngay cả khi dùng mã hóa đối xứng để encrypt và decrypt thông tin truyền và nhận thì attacker vẫn có thể tóm được key và đọc được thông tin như thường .Vì vậy điểm yếu đó đã được khắc chế trong mã hóa bất đối xứng .Ý tưởng là thay vì gửi khóa cho phía client thì server sẽ gửi ổ khóa và để client khóa thông điệp trong một chiếc hộp mà chỉ có server mới có thể giải mã được .Cho nên các client sẽ không đọc được thông điệp của nhau, và chỉ có server với private key mới mở khóa được những chiếc hộp này .Thực tế thì public key vừa dùng để mã hóa vừa dùng để giải mã thông tin nhận và gửi lên server.

- **Hạn chế**

- Một nhược điểm của mã hóa bất đối xứng đó là tốc độ giải mã chậm hơn so với mã hóa đối xứng, tức là chúng ta phải tốn nhiều năng lực xử lý của CPU hơn, phải chờ lâu hơn, dẫn đến "chi phí" cao hơn. Khoảng thời gian lâu hơn là bao nhiêu thì còn tùy vào thuật toán mã hóa, cách thức mã hóa và key.
- Vấn đề trao đổi khóa giữa người gửi và người nhận: Phải truyền khóa trên kênh an toàn để giữ bí mật. Ngay nay điều này tỏ ra không hợp lý vì khối lượng thông tin luân chuyển trên khắp thế giới là rất lớn.
- Tính bí mật của khóa (Tính không từ chối): Vì khóa 2 người dùng chung nên khi khóa bị lộ không có cơ sở quy trách nhiệm cho ai.

2.5.5.5 Viết ứng dụng bằng ngôn ngữ C#

a. Ngôn ngữ C# là gì ?

- 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. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
- Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.Trong các ứng dụng sử dụng .NET Framework,

mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

- Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

b. Lý do lựa chọn C#

Là ngôn ngữ đơn giản

- C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).
- Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

Là ngôn ngữ hiện đại

- Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới học lập trình có thể chúng ta sẽ cảm thấy những đặc tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các nội dung khoá học này.

Là ngôn ngữ lập trình hướng đối tượng

- Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa

(inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

Là một ngôn ngữ ít từ khóa

- C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

c. Chương trình ứng dụng ngôn ngữ C#

File chương trình đính kèm trong báo cáo bài tập lớn

Chương 3. Kiến thức lĩnh hội và bài học kinh nghiệm

3.1 Nội dung đã thực hiện

Những kiến thức và kĩ năng sau khi thông qua bài tập lớn mà nhóm thực hiện bài tập lớn đã tiếp thu được:

- Định nghĩa về chữ ký điện tử và cụ thể về chữ ký điện tử RSA
- Nắm vững các nội dung của thuật toán hệ mã RSA
- Định nghĩa về hàm băm SHA và cách sử dụng hàm băm
- Định nghĩa và nội dung của mã hóa bất đối xứng
- Nắm vững ngôn ngữ lập trình PHP, Python, C# và Java để thực hiện áp dụng xây dựng chương trình ứng dụng demo cho chữ lí điện tử
- Các công cụ để thực hiện demo chương trình như Visual Studio Code, NetBeans…

- Những bài học kinh nghiệm được rút ra sau khi kết thúc bài tập lớn là rất nhiều và sâu sắc
- Đề tài nghiên cứu về chữ kí điện tử sử dụng thuật toán RSA và viết ví dụ minh họa, trong đó hệ mã RSA được tập trung tìm hiểu bao gồm cách thức hoạt động, quy trình kí, quy trình kiểm tra chữ kí. Bên cạnh đó, nghiên cứu về phương pháp mã hóa bất đối xứng ứng

dụng trong chữ kí điện tử, đưa ra được những ưu điểm, nhược điểm của phương pháp khi ứng dụng vào bài toán.

- Một mặt quan trọng khác là hàm băm(MD5), chúng em đã được lĩnh hội thêm kiếm thức về các thuật toán không sử đụng khóa để mã hóa, tìm hiểu được tính chất của hàm băm và cách thức hoạt động của nó.
- Từ đó có được những kiến thức cần thiết để xây dựng chương trình ứng dụng chữ kí điện tử đã đạt được một số kết quả như sau: Giới thiệu một cách khái quát ứng dụng của thuật toán RSA, phương pháp mã hóa bất đối xứng, kiến thức về hàm băm và cài đặt chương trình ứng dụng. "Chữ ký số và ứng dụng trong giao dịch hành chính điện tử " để thực hiện các quá trình: Ký và xác thực chữ ký, mã hóa và giải mã các tập tin… giao dịch qua mạng.

3.2 Xây dựng hướng phát triển đề tài

Do thời gian nghiên cứu có hạn, nên chương trình mới chỉ mô phỏng được các thao tác: ký, xác thực chữ ký, mã hóa và giải mã tập tin mà chưa thiết kế một cách hoàn chỉnh để có thể kết nối trực tiếp vào một số phần mềm : gửi nhận email, phần mềm quản lý văn bản … Hướng phát triển của đề tài là xây dựng chương trình để có thể kết nối trực tiếp vào một số phần mềm gửi nhận email và phần mềm quản lý văn bản. Đồng thời xây dựng một hệ thống chứng thực khóa công khai cho các thành viên, nhằm tránh trường hợp bị người khác giả mạo khóa công khai của người nhận khi thực hiện trao đổi thông tin.

Cuối cùng, với những kết quả đạt được của đề tài nghiên cứu, tuy còn có những hạn chế, nhưng đã giúp chúng em có được khả năng nghiên cứu cơ bản về bảo mật và xác thực thông tin. Từ đó có thể xây dựng các ứng dụng về bảo mật và xác thực thông tin ở những cấp độ an toàn khác nhau.