

# BÁO CÁO

**MÔN:** Project I-Bộ môn Khoa học máy tính

**Đề tài:** Ứng dụng giải pháp chữ ký điện tử trong trao đổi thông tin bằng phương pháp kết hợp băm MD5 và mã hóa khóa công khai RSA

*Sinh viên thực hiện:* **Phùng Ngọc Duy**

*Mã số sv:* 20101256

*Email:* tonybka@gmail.com

*Giáo viên hướng dẫn:* **TS. Trần Vĩnh Đức**

*Hà Nội, tháng 11 năm 2012*

## Mục lục

### Chương I. Tổng quan

1. Lý do chọn đề tài
2. Mục đích nghiên cứu
3. Nhiệm vụ nghiên cứu
4. Phương pháp nghiên cứu

### Chương II. Nội dung

1. An toàn thông tin, chứng thực thông tin và chữ kí điện tử
2. Mã hóa dữ liệu và giải mã
  - a. Tổng quan về mã hóa dữ liệu
  - b. Mã hóa đối xứng
  - c. Mã hóa bất đối xứng
  - d. Hàm băm
  - e. Mã hóa RSA và sự chuẩn bị toán học cần thiết
  - f. Chứng nhận số (Digital certificate) và tổ chức chứng nhận số (Certificate Authority)
3. Chữ kí điện tử
  - a. Tổng quan về chữ kí điện tử
  - b. Kết hợp hàm băm vào chữ kí điện tử
  - c. Sử dụng chữ kí điện tử

### Chương III. Cài đặt

1. Cài đặt chương trình tạo chữ kí điện tử với kết hợp của RSA và MD5
  - a. Môi trường và công cụ cài đặt
  - b. Tích hợp thư viện Crypto++ vào môi trường lập trình
  - c. Mục đích hướng tới
2. Kết quả đạt được
3. Hướng phát triển

## Chương I. Tổng quan

### 1. Lý do chọn đề tài

Ngày nay công nghệ thông tin ngày càng có vai trò quan trọng trong các thành phần của xã hội như kinh tế, chính trị, quân sự.... Với một chức năng không thể thiếu và được ứng dụng mạnh mẽ là truyền thông. Rất nhiều thông tin liên quan đến công việc hằng ngày đều di máy tính quản lý và được trao đổi với nhau qua mạng internet. Vấn đề nảy sinh ra là làm thế nào để chứng thực được chính xác nguồn tin và thông tin được nhận có phải là do người đó gửi và thông tin nhận được đã bị thay đổi thay chưa?

Khi một thông tin gửi từ người A tới B đã bị thay đổi do hacker hoặc là virus thì sau khi nhận được thông tin, người B không có phương pháp để có thể kiểm tra được nội dung thông tin đó có phải là của người A gửi hay không cũng như không đảm bảo được tính toàn vẹn của thông tin đó.

Vấn đề nêu trên đã đưa chúng ta tới giải pháp “**Sử dụng chữ ký điện tử trong trao đổi thông tin**” để nâng cao khả năng bảo mật thông tin khi truyền tải thông tin qua internet.

### 2. Mục đích nghiên cứu

Tìm hiểu lý thuyết về chứng thực thông tin, chữ ký điện tử, các thuật toán cần thiết trong quá trình mã hóa. Cài đặt chương trình minh họa chữ ký sử dụng chữ ký điện tử bằng phương pháp kết hợp hàm băm MD5 và thuật toán mã hóa RSA

### 3. Nhiệm vụ nghiên cứu

- Nắm được lý thuyết chứng thực thông tin
- Quy trình sử dụng chữ ký điện tử
- Tìm hiểu phương thức mã hóa dữ liệu cơ bản
- Thuật toán mã hóa RSA và các chuẩn bị toán học
- Hàm băm
- Cài đặt chương trình đồ họa minh họa cho những gì tìm hiểu được

### 4. Phương pháp nghiên cứu

- Tìm hiểu lịch sử phát triển của mã hóa thông tin, lý thuyết mã hóa thông tin và quy trình sử dụng chữ ký điện tử từ những tài liệu có trên internet
- Hàng tuần có 1 buổi trao đổi với giảng viên hướng dẫn, làm theo những định hướng mà thầy đề ra để có thể áp dụng những lý thuyết tìm hiểu được theo hướng đi chính xác.

## Chương II. Nội dung

### 1. An toàn thông tin, chứng thực thông tin và chữ ký điện tử

Xã hội đang trong giai đoạn phát triển mạnh mẽ, trong đó nhu cầu trao đổi thông tin là không thể thiếu và quan trọng đối với tất cả các lĩnh vực của đời sống. Cùng với sự phát triển đó đòi hỏi tính bảo mật và chứng thực thông tin ngày càng có vai trò quan trọng phổ biến. Có nhiều trường hợp khác nhau về nhu cầu an toàn thông tin cùng với các khái niệm:

**Bảo mật:** Giữ thông tin được giữ bí mật với tất cả mọi người ngoại trừ những trường hợp , cá nhân đặc biệt.

**Toàn vẹn thông tin:** Đảm bảo thông tin không bị thay đổi bởi cá nhân khác không có thẩm quyền trong quá trình truyền tải trên internet.

**Chữ ký:** một cách gắn kết một gói thông tin với một thực thể để đảm bảo tính cá nhân của người gửi.

**Cấp chứng chỉ:** Cấp một sự xác nhận thông tin bởi một thực thể được tín nhiệm.

Để xác thực một người và mức độ tin cậy của thông tin trên máy tính ta có thể sử dụng các phương pháp như: thẻ bảo mật, password và chữ ký điện tử.

## 2. Mã hóa dữ liệu và giải mã

### a. Tổng quan về mã hóa dữ liệu

Sự phát triển nhanh chóng của Internet đã có nhiều đến công việc kinh doanh và người tiêu dùng làm thay đổi cách con người sống và làm việc. Tuy nhiên , đi cùng với sự phát triển đó là nhu cầu cần nâng cao tính bảo mật trong truyền tải thông tin , đặc biệt đối với các thông tin nhạy cảm được truyền tải trên mạng.

Mã hóa là lĩnh vực nghiên cứu các phương thức, thuật toán để bảo mật thông tin. Các sản phẩm của lĩnh vực này là các hệ mật mã, hàm băm, các cơ chế phân phối , quản lý khóa và giao thức mật mã.

Quá trình mã hóa trên máy tính dựa trên khoa học về mật mã(Cryptography) đã được con người sử dụng từ rất lâu, trong đó sử dụng nhiều nhất có lẽ là chính phủ, chủ yếu trong lĩnh vực quân sự.. Hầu hết các mã hóa hiện nay đều dựa vào máy tính do các mã hóa do con người tự sinh ra rất dễ bị phá bởi máy tính, do máy tính có khả năng tính toán tốt hơn con người. Như vậy , máy tính tiến hành mã hóa dựa trên những thuật toán mã hóa mà con người tạo ra.

Các hệ thống mã hóa trên máy tính thông thường chia làm hai loại ( cách phân loại này dựa vào đặc điểm của khóa sử dụng để mã hóa):

- Mã hóa với khóa bí mật ( *Private key Encryption*)
- Mã hóa khóa công khai ( *Public key Encryption*)

Nếu dựa theo phương pháp mã hóa thì mã hóa thông tin được chia thành:

- Mã hóa cổ điển(Classical Cryptography)
- Mã hóa đối xứng(Symetric Cryptography)
- Mã hóa phi đối xứng (Asymetric Cryptography)
- Hàm băm ( Hash Function)

### b. Mã hóa đối xứng

Là lớp các thuật toán mã hóa mà trong đó các khóa dùng cho việc mã hóa và giải mã có quan hệ rõ ràng với nhau( có thể dễ dàng tìm thấy khóa này nếu biết khóa kia).

Khóa dùng để mã hóa có liên hệ rõ ràng với khóa để giải mã có nghĩa chúng có thể hoàn toàn giống nhau, hoặc chỉ khác nhau nhờ một biến đổi đơn giản giữa hai khóa.

Phương thức mã hóa đối xứng thường nhanh hơn nhiều mã hóa bất đối xứng nên trên thực tế, các khóa này đại diện cho **một** bí mật được hưởng bởi hai bên và được sử dụng để giữ gìn bí mật trong kênh truyền thông tin. Nếu key này bị mất hay bị lộ, khi đó sẽ không đảm bảo tính bảo mật dữ liệu nữa.

Mã hóa đối xứng được chia làm hai loại:

- Block cipher  
Là một giải pháp hoạt động chống lại sự hạn chế của dữ liệu tĩnh. Dữ liệu được chia ra thành các blocks với các size cụ thể và mỗi blocks được mã hóa một cách khác nhau.
- Stream cipher  
Là giải pháp hoạt động chống lại dữ liệu luôn luôn sử dụng một phương thức để truyền. Một vùng đệm, ít nhất bằng một block, đợi cho toàn bộ thông tin của block đó được chứa trong vùng đệm sau đó block đó sẽ được mã hóa rồi truyền cho người nhận. Một sự khác nhau cơ bản giữa dữ liệu được truyền và dữ liệu nguyên bản. Không như giải pháp sử dụng mật mã đối xứng là mỗi block được sử dụng một key khác nhau trong quá trình truyền thông tin.

### c. Mã hóa bất đối xứng

Mã hóa bất đối xứng là phương thức mã hóa mà key dùng để mã hóa và key dùng giải mã hoàn toàn khác nhau và từ key này không thể có được key kia. Mật mã bất đối xứng sử dụng một cặp key đó là public key và private key, trong quá trình truyền thông tin sử dụng mật mã bất đối xứng chúng cần một cặp key duy nhất. Nó tạo ra khả năng sử dụng linh hoạt và phát triển trong tương lai hơn là mật mã đối xứng. Private key cần phải giữ riêng và đảm bảo tính bảo mật và nó không được truyền trên mạng, public key được cung cấp miễn phí và được chia sẻ cho mọi người.

Ví dụ, khi Tom muốn gửi thông tin cho Bob, Tom sử dụng public key của Bob để mã hóa thông tin rồi gửi cho Bob. Khi Bob nhận thông tin đã được mã hóa của Tom sẽ giải mã thông tin bằng Private key của mình.

Mật mã bất đối xứng hoạt động chậm hơn phương thức mật mã đối xứng, không phải nó mã hoá một khối lượng dữ liệu lớn. Nó thường được sử dụng để bảo mật quá trình truyền key của mật mã đối xứng. Nó cung cấp bảo mật cho quá trình truyền thông tin bằng các dịch vụ: Authentication, Integrity, Protection, và Nonrepudiation.

Một số phương thức mã hóa bất đối xứng :

- Rivest Shamir Adleman(RSA)
- Error Correcting
- Diffie- Hellman

### d. Hàm băm

Hàm băm là chuyển đổi một thông điệp có độ dài bất kì thành một dãy bit có độ dài cố định. Các hàm băm nhận một chuỗi bit có chiều dài tùy ý( hữu hạn) làm dữ liệu đầu vào và tạo ra một chuỗi bit mới có chiều dài cố định  $n$  bit(  $n > 0$ ) gọi là mã băm.

**Tính chất cơ bản của hàm băm:**

- Tính tiền ảnh: với mọi đầu ra  $y$  cho trước không thể tính toán để tìm được bất kỳ dữ liệu đầu vào  $x'$  nào sao cho giá trị băm  $h(x')$  bằng giá trị đầu ra  $y$  đã cho.
- Tính tiền ảnh thứ hai: với mọi dữ liệu đầu vào  $x_1$  cho trước, không thể tính toán để tìm ra được bất kỳ một đầu vào  $x_2$  nào ( $x_1 \neq x_2$ ) sao cho giá trị băm  $h(x_1) = h(x_2)$ .
- Tính kháng xung đột: Không thể tính toán để tìm được hai dữ liệu đầu vào  $x_1 \neq x_2$  sao cho chúng có cùng giá trị băm.

Trong lĩnh vực mã hóa thông tin, mã băm được xem như hình ảnh đặc trưng thu gọn của mỗi chuỗi bit có độ dài tùy ý hữu hạn, và được dùng để nhận diện chuỗi bit đó. Kết hợp với các công cụ tạo chữ ký số, các hàm băm được dùng cho việc đảm bảo tính toàn vẹn dữ liệu.

- Về **MD5(Message Digest)**

Ronald Rivest là người đã phát minh ra các hàm Băm **MD2**, **MD4** (1990) và **MD5** (1991). Do tính chất tương tự của các hàm Băm này, sau đây chúng ta sẽ xem xét hàm Băm **MD5**, đây là một cải tiến của **MD4** và là hàm Băm được sử dụng rộng rãi nhất, nguyên tắc thiết kế của hàm băm này cũng là nguyên tắc chung cho rất nhiều các hàm băm khác.

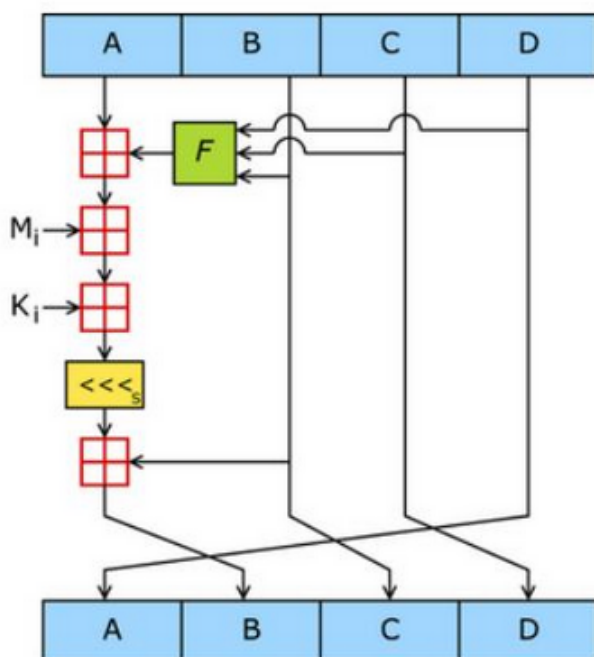
**Thuật mã hóa:**

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

Hoạt động của bộ đệm: Trên 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 bits. Phần còn lại sẽ được lấp đầy bởi 1 số nguyên 64 bits 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 bits, kí hiệu là A,B,C và D. Các giá trị này là các hằng số cố định. Sau đó thuật toán sẽ luân phiên hoạt động trên các khối 512 bits. Mỗi khối sẽ phối hợp với 1 bộ. Quá trình xử lý một khối thông điệp bao gồm 4 bước tương tự nhau, gọi là vòng. Mỗi vòng lại gồm 16 quá trình tương tự nhau dựa trên hàm một chiều F, phép cộng module và phép xoay trái.

**Một quá trình trong một vòng được mô tả theo hình sau:**



### e. Mã hóa RSA và sự chuẩn bị toán học cần thiết

#### - Thuật toán mã hóa RSA

Thuật toán mã hóa RSA có hai khóa: Khóa công khai và khóa bí mật. Mỗi khóa là những số cố định chỉ sử dụng trong quá trình mã hóa và giải mã.

Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người có thể mã hóa nhưng chỉ có một người biết khóa cá nhân mới có thể giải mã được.

Mô phỏng trực quan:

Bob một gửi cho Tom một thông điệp mật mà muốn chỉ có Tom biết. Để làm được điều này Tom được cho Bob một chiếc hộp có khóa mở sẵn và giữ lại chìa khóa. Sau khi nhận được chiếc hộp Bob bỏ thông điệp vào đó rồi sập khóa lại (Với loại khóa sập thông thường, sau khi chốt khóa thì kể cả Bob cũng không thể đọc lại thông điệp đó nữa). Sau đó Bob gửi chiếc hộp cho Tom, Tom mở chiếc hộp với khóa tương ứng của nó và đọc thông điệp được đựng trong đó. Trong ví dụ này, chiếc hộp với khóa được mở sẵn đóng vai trò là khóa công khai, chiếc chìa khóa được Tom giữ lại là khóa bí mật mà chỉ mình Tom có.

#### - Các khái niệm toán học cần thiết

+ Hàm một phía:

Hàm một phía là hàm dễ tính toán ra quan hệ một chiều nhưng rất khó để có thể tính ngược lại (biết  $x$  có thể tính ra  $y$  nhưng nếu biết  $y$  rất khó có thể suy ra được  $x$ ), “khó” ở đây có nghĩa là vẫn có thể tính ra được nhưng phải mất một khoảng thời gian vô tận để tính ra nó.

Hộp thư là 1 ví dụ thực tế về hàm một phía, bất kỳ ai cũng có thể bỏ thư vào thùng như một hành động công cộng nhưng mở thùng thư không phải là một hành động công cộng, muốn mở nó ta cần có chìa khóa của hộp thư.

**+ Số nguyên tố**

Khái niệm số nguyên tố là một khái niệm toán học cơ bản để chỉ những số nguyên chỉ chia hết được cho 1 và chính nó

**+Khái niệm nguyên tố cùng nhau(relatively prime)**

Trong toán học hai số nguyên a và b được gọi là nguyên tố cùng nhau nếu chúng có ước số chung lớn nhất là 1

**+Khái niệm Modulo**

Với m là một số nguyên dương . Ta nói hai số nguyên a và b là đồng dư với nhau modulo m nếu m chia hết cho hiệu a-b ( viết là  $m|(a-b)$  ).

Kí hiệu  $a \equiv b \pmod{m}$ .

Ta có  $a \equiv b \pmod{m}$  khi và chỉ khi tồn tại số nguyên k sao cho  $a=b+km$

**+Phi hàm Euler**

Theo lý thuyết số thì hàm số Euler( ký hiệu  $\varphi(n)$  ) của một số nguyên dương n được định nghĩa là số các số nguyên dương nhỏ hơn hoặc bằng n nguyên tố cùng nhau với n.

Ví dụ :  $\varphi(5)=4$ ,  $\varphi(9)=6$ .

**Từ những lý thuyết toán học chuẩn bị ở trên ta có thể tiến hành tạo khóa như sau :**

- Tạo ngẫu nhiên hai số nguyên tố vô cùng lớn p và q
- Tính  $N=p.q$
- Tính giá trị hàm số Euler  $\varphi(n)=(p-1)(q-1)$
- Chọn một số tự nhiên e sao cho  $1 < e < \varphi(n)$  và là số nguyên tố cùng nhau với  $\varphi(n)$
- Tính d sao cho  $de \equiv 1 \pmod{\varphi(n)}$

Vậy hệ thống khóa gồm: - Khóa công khai: n,e

- Khóa bí mật: n,d

Công thức mã hóa:  $c=m^d \pmod{n}$ .

m: đoạn mã có được sau khi bấm thông điệp muốn gửi

c: thông điệp sau khi đã tiến hành bấm và mã hóa RSA

Công thức giải mã:  $m=c^e \pmod{n}$ .

**f. Chứng nhận số ( Digital certificate) và tổ chức chứng nhận số ( Certificate Authority)**

Ví dụ A muốn gửi thông điệp cho B và mã hóa theo phương pháp khó công khai . Lúc này A cần phải mã hóa thông điệp bằng public key của B. Trong trường hợp public key bị giả mạo, tin tức sẽ tự sinh ra một cặp khóa public key private key,sau đó đưa cho A khóa public key và nói rằng đây là khóa của B.



Nếu A dùng khóa này để mã hóa thông tin muốn gửi cho B thì mọi thông tin đó truyền đi sẽ bị tin tặc đọc được.

Vấn đề được giải quyết nếu có một bên thứ ba được tin cậy, gọi là CA chứng nhận public key. Những public key đã được CA chứng nhận gọi là chứng nhận điện tử( digital certificate).

Một chứng nhận điện tử có thể được xem như là một “ chứng minh thư “. Nó được các tổ chức tin cậy tạo ra. Tổ chức này được gọi là tổ chức chứng nhận khóa công khai Certificate Authority (CA). Một khi public key được tổ chức CA chứng nhận thì có thể dùng khóa đó để trao đổi dữ liệu trên mạng với mức độ bảo mật cao.

Cấu trúc của một chứng nhận điện tử bao gồm:

- Tên CA tạo ra chứng nhận
- Ngày hết hạn của chứng nhận
- Thông tin về thực thể được chứng nhận
- Khóa công khai được chứng nhận
- Chữ kí do khóa private key của CA tạo ra và đảm bảo giá trị chứng nhận

### 3. Chữ ký điện tử ( Electronic Signature)

#### *a. Tổng quan về chữ ký điện tử*

Chữ ký điện tử là thông tin đi kèm theo dữ liệu nhằm mục đích xác định người chủ sở hữu dữ liệu đó. Chữ ký điện tử được sử dụng trong các giao dịch điện tử. Dựa trên thực tế, chữ kí điện tử cần đảm bảo các chức năng : xác định được người chủ của dữ liệu nào đó : văn bản,tài liệu..... , dữ liệu đó có bị thay đổi hay không.

***Khái niệm chữ ký điện tử và chữ ký số có thể được dùng thay thế cho nhau mặc dù chúng không hoàn toàn giống nhau.***

#### *b. Kết hợp hàm băm vào chữ ký điện tử*

Hàm băm trợ giúp cho các sơ đồ ký số nhằm giảm dung lượng của dữ liệu cần thiết để truyền qua mạng

Hàm băm thường được kết hợp với chữ ký số để tạo ra một loại chữ ký điện tử an toàn hơn( không thể sao chép)đồng thời có thể kiểm tra được tính toàn vẹn của dữ liệu truyền tải.

Hàm băm được ứng dụng rất nhiều trong an toàn trao đổi thông tin vì tính ngắn gọn , giúp người dùng dễ dàng nhận ra rằng thông tin của mình đã bị thay đổi hay chưa từ đó giữ được sự tin tưởng của người sử dụng

#### *c. Sử dụng chữ ký điện tử (kết hợp giữa MD5 & RSA)*

- **Tạo chữ ký số :**

Sử dụng ứng dụng tạo chữ ký số từ khóa bí mật, khóa bí mật do nhà cung cấp dịch vụ chứng thực

Để an toàn và chống copy khóa bí mật một số nhà cung cấp dịch vụ lưu trữ khóa bí mật trong thiết bị phần cứng thông dụng, 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 hoại.

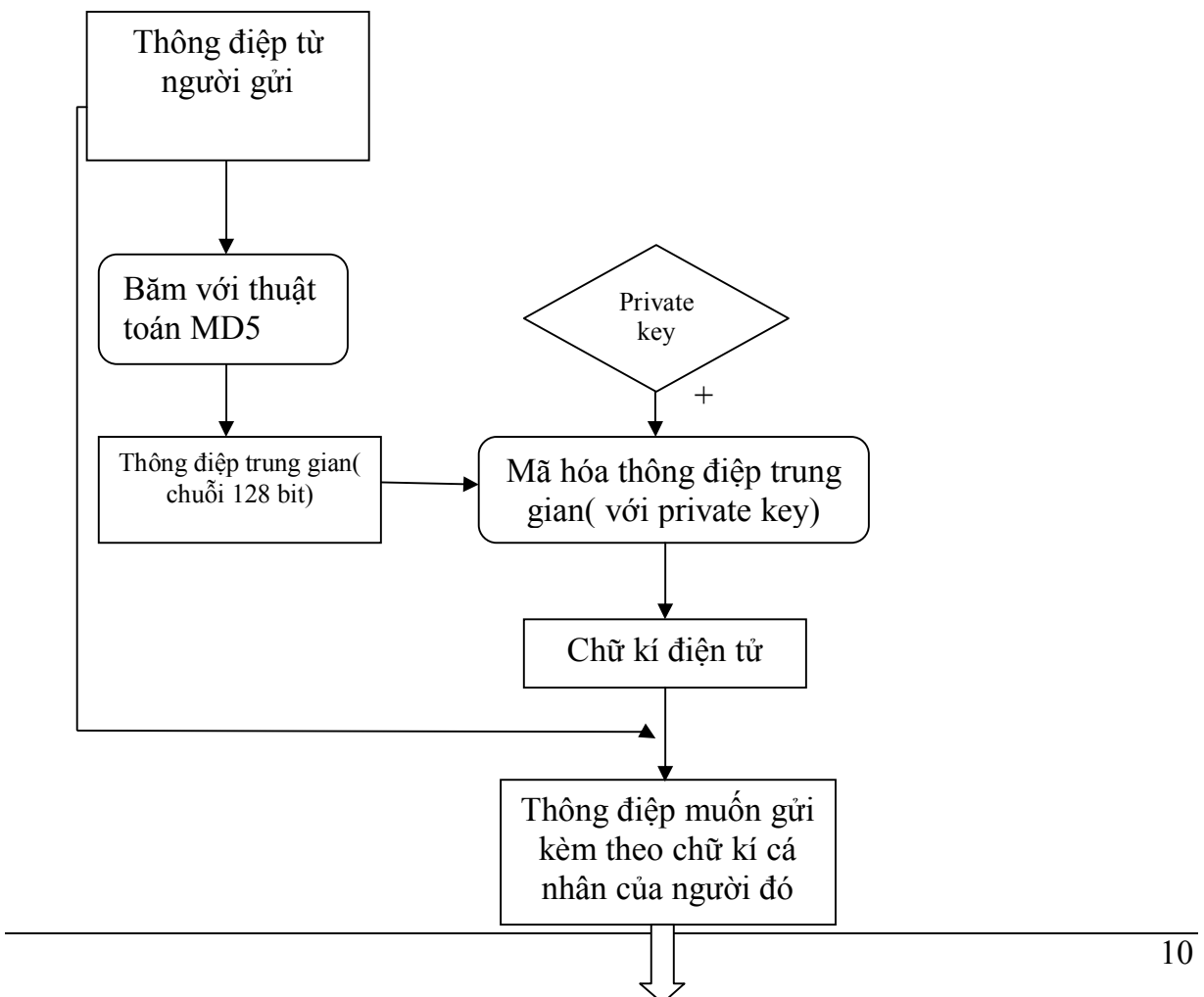
**Dưới góc nhìn của người lập trình:** Người dùng sử dụng hàm băm để thu gọn thông điệp cần gửi tiếp theo mã hóa thông điệp thu gọn đó với RSA, kết quả thu được là chữ ký số. Cuối cùng, gửi đính kèm thông điệp muốn gửi với chữ ký số tới người nhận.

**- Kiểm tra chữ ký số :**

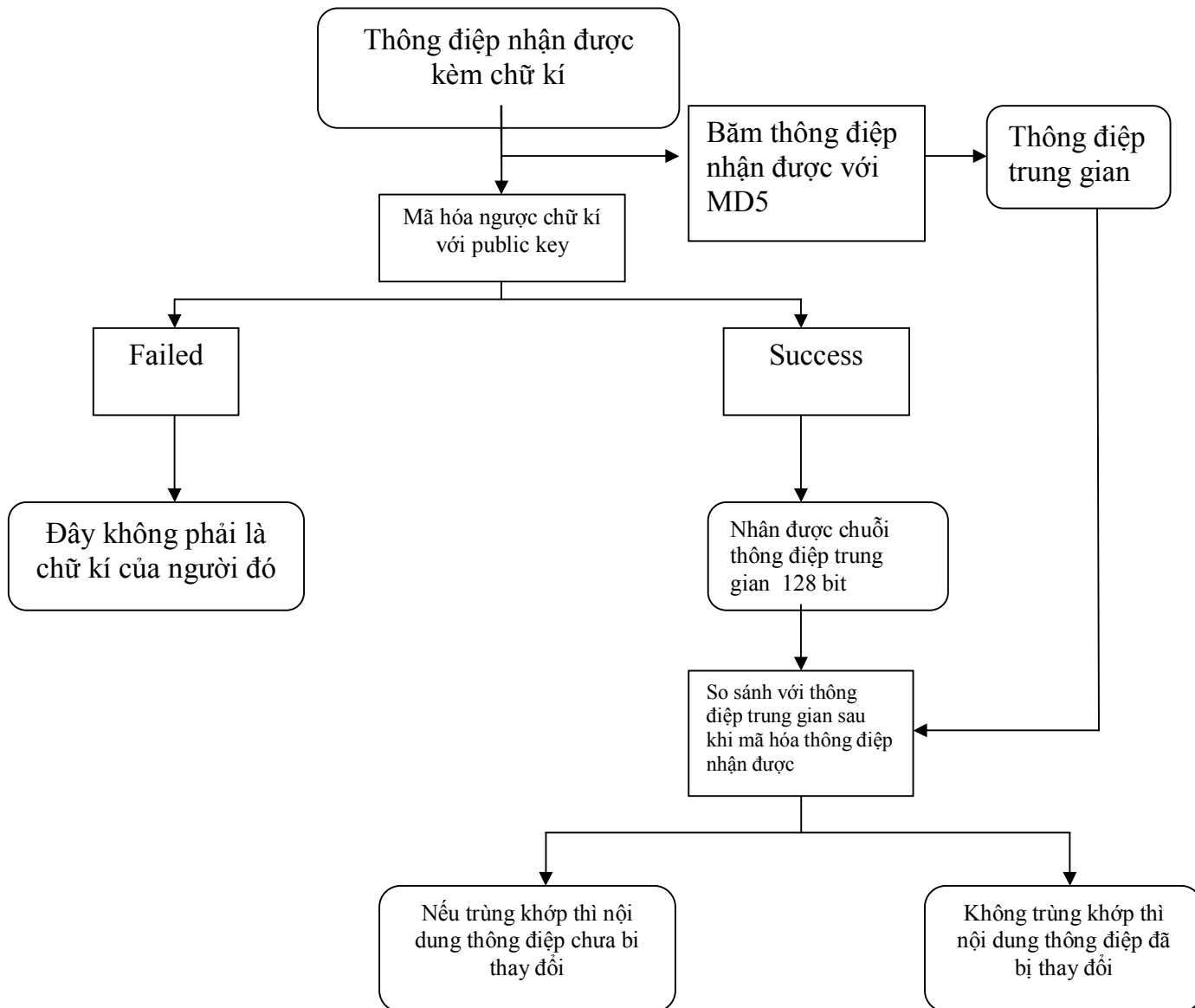
Khi giao dịch điện tử người nhận phải kiểm tra được tính pháp lý của chữ ký số của người giao dịch với mình gửi đến.

Việc kiểm tra là so sánh tính đồng nhất của chuỗi thu gọn sau khi sử dụng hàm băm (cùng phương pháp băm của người gửi) để băm thông điệp nhận được với chuỗi thu gọn nhận được sau khi giải mã chữ ký với khóa công khai của người gửi.

**Sơ đồ tạo chữ kí điện tử**



### Sơ đồ chứng thực chữ kí:



## Chương III. Cài đặt

### 1. Cài đặt chương trình tạo chữ ký điện tử với sự kết hợp của MD5 và RSA

#### a. Môi trường và công cụ cài đặt

Để cài đặt chương trình ta sử dụng công cụ lập trình là Visual Studio 2008 với ngôn ngữ lập trình C++ sử dụng bộ thư viện MFC(Microsoft Foundation Classes) và bộ thư viện Crypto++ 5.6.0

( *Opensource C++ class library of cryptographic algorithms and schemes* written by **Wei Dai**)

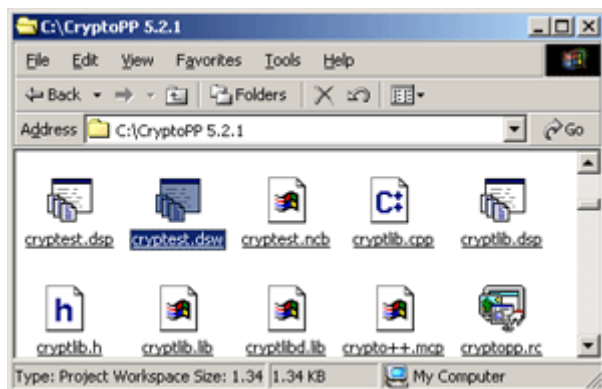
- Chúng ta có thể cập nhật thư viện Crypto++ update mới nhất tại địa chỉ

<http://www.cryptopp.com/>

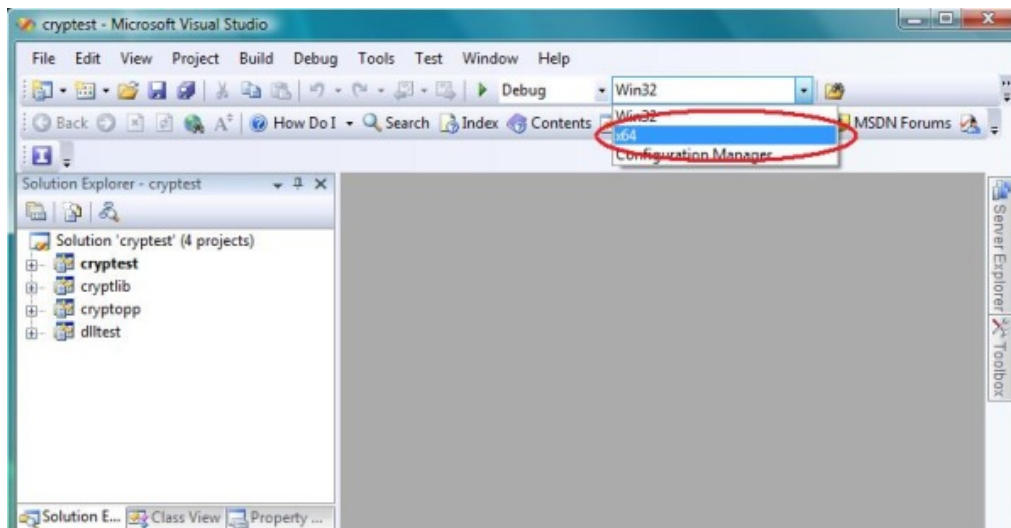
#### b. Tích hợp thư viện Crypto++ vào môi trường lập trình

Tải bộ thư viện Crypto++ từ trên trang chủ rồi giải nén, để tránh mất liên kết giữa thư viện sau khi tạo liên kết với công cụ lập trình chúng ta nên giải nén file vừa tải được vào ổ C:\ ( ổ đĩa cài đặt Windows) và được thư mục C:\CryptoPP 5.6.0\.

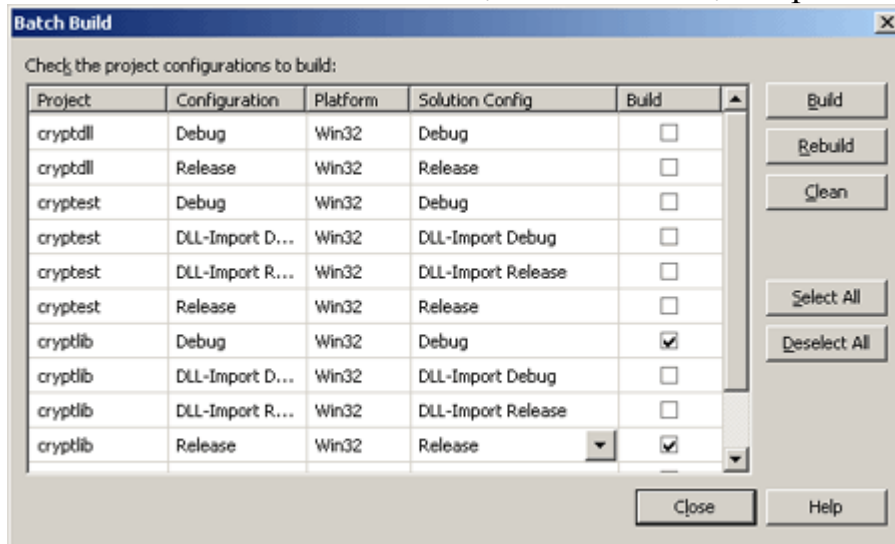
Khởi động Visual Studio rồi Open Project, với cửa sổ Open của Visual tìm đến file **crypttest.dsw** hoặc **crypttest.sln** rồi chọn để mới nó, như vậy cả bộ thư viện sẽ được load lên Visual gồm các file .cpp và file .h .



Như mặc định thì 32 bit nhị phân sẽ được build để build thư viện với 64bit nhị phân thì ra chọn Configuration Manager và chỉnh sửa lại



Trên thanh Menu của Visual chọn **Build** rồi chọn tiếp **Batch Build...**

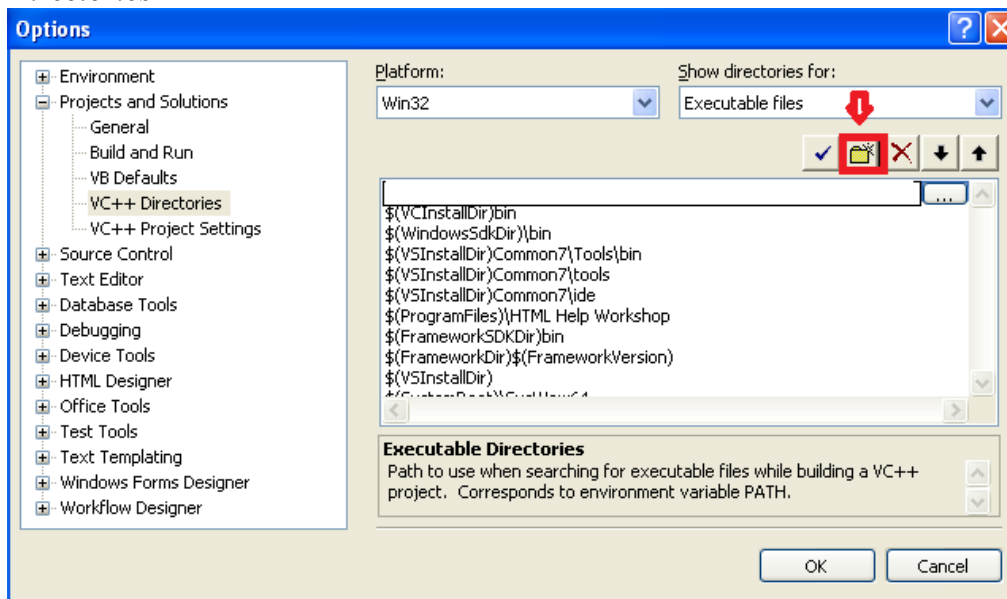


Trong cửa sổ hiện lên ta tích dấu vào những thành phần như cửa sổ trên rồi nhấn **Build**

Đợi một lúc để Visual thực hiện quá trình build thư viện, nếu sau quá trình đó mà dưới cửa sổ **Output** hiện thị nội dung : *1 succeedd, 0 failed...* thì quá trình tạo liên kết của chúng ta sắp hoàn thành rồi.

Tiếp theo ta tới thư mục giải nén thư viện Crypto++ ( *C:\CryptoPP 5.6.0\*) mở thư mục *C:\CryptoPP 5.6.0\Debug* rồi đổi tên file *cryptlib.lib* thành *cryptlibd.lib* , copy file này với file *cryptlib.lib* trong thư mục *C:\CryptoPP 5.6.0\Release* ( không đổi tên) ra ngoài thư mục *C:\CryptoPP5.6.0\*

Add vị trí của những file header (.h) và source files vào môi trường Visual Studio: - Chọn Tool|Option , chọn Tab *Projects and Solutions*, chọn Tab *VC++ Directories*

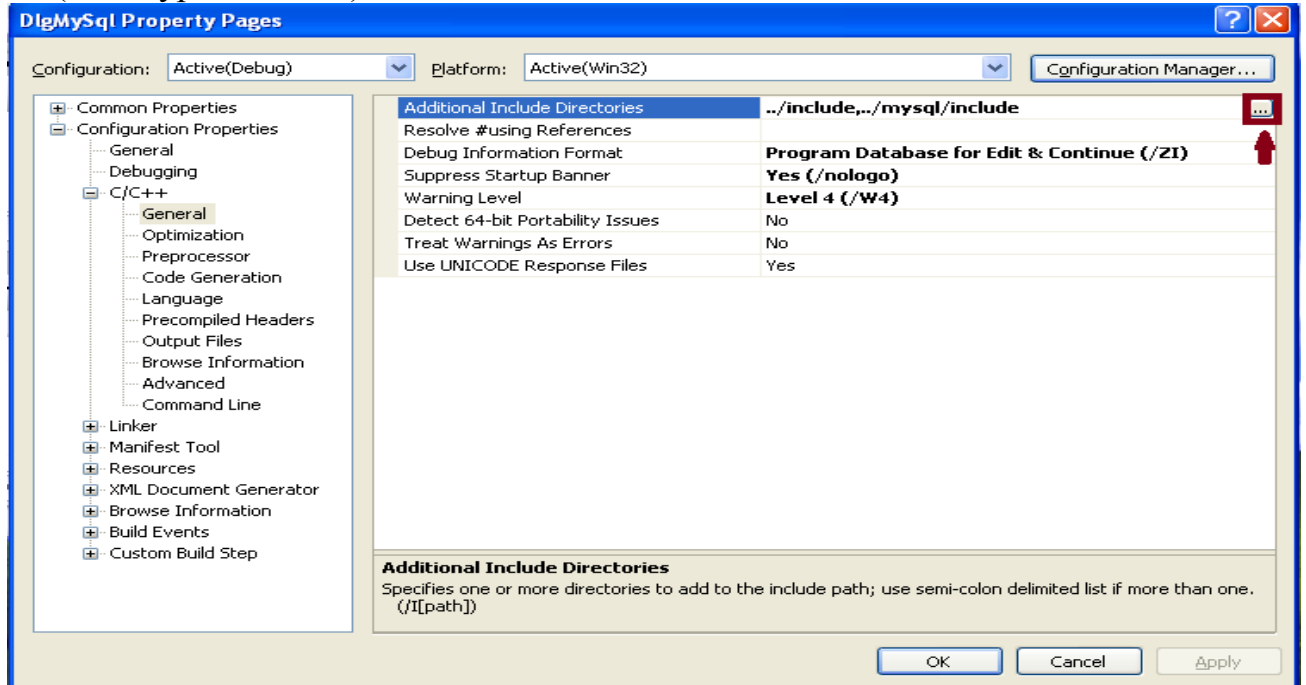


Hình 1

Ấn vào biểu tượng thư mục như trên hình 1 rồi Add tên đường dẫn thư mục chứa thư viện (Trong trường hợp này là *C:\CryptoPP 5.6.0*) vào ô bên dưới và OK để kết thúc.

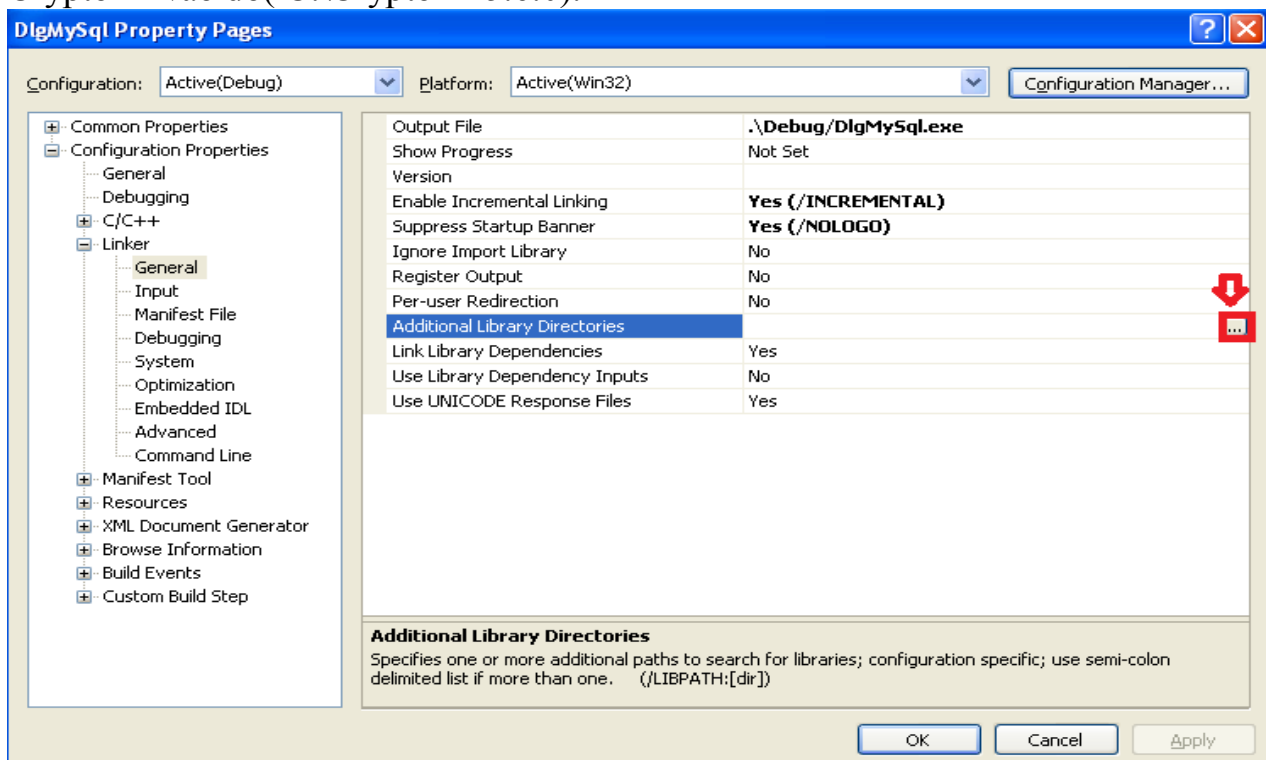
Tại mỗi Project muốn sử dụng thư viện Crypto++ ta phải config lại bằng cách sau:

Sau khi mở Project , trên thanh Menu của Visual ta chọn Project|Properties, chọn Tab Configuration Properties, ấn vào C/C++|General , trên mục Additional Include Directories (Hình 2) ta thêm tên thư mục chứa thư viện Crypto++ vào đó ( C:\CryptoPP 5.6.0).



Hình 2

Tiếp tục, Vẫn ở Tab Configuration Properties t chọn Linker|General rồi tại mục Additional Library Directories ( như hình 3) ta Add tên đường dẫn thư mục chứa Crypto++ vào đó( C:\CryptoPP 5.6.0).



Hình 3

Cuối cùng tại mỗi file header (.h) của một class có sử dụng thư viện Crypto++ ta thêm đoạn code sau để tạo liên kết thực sự tới thư viện:

```
#ifdef _DEBUG
# pragma comment ( lib, "cryptlibd" )
#else
# pragma comment ( lib, "cryptlib" )
#endif
```

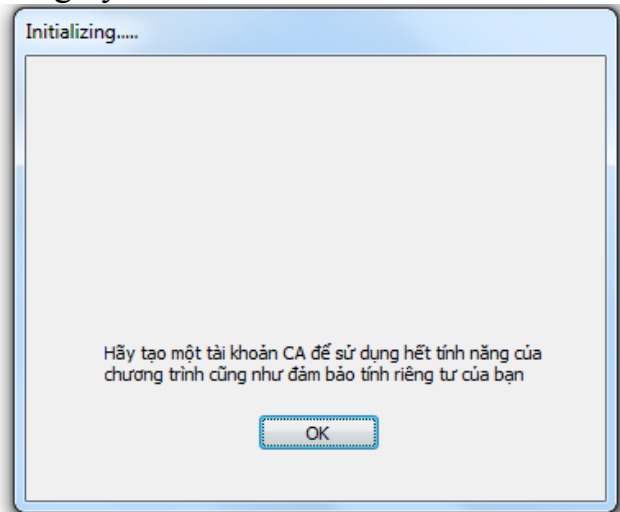
Vậy là chúng ta đã có thể sử dụng thư viện 1 cách hoàn hảo!.

### c. Mục đích hướng tới

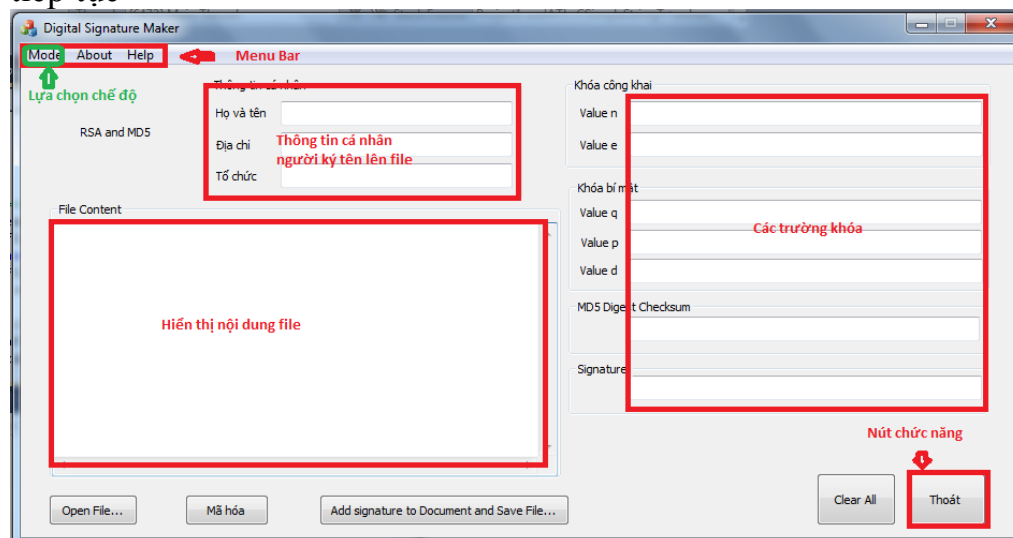
Cài đặt thành công một chương trình tạo chữ ký điện tử và có khả năng quản lý dữ liệu đăng ký của người dùng chức năng Certificate Authority.

### 2.Kết quả đạt được

Chương trình hoàn thành chức năng mã hóa file , cho phép đăng ký tài khoản Certificate và lưu trữ lại cho lần sử dụng tiếp theo có thể sử dụng lại thông số đã đăng ký.



Màn hình khởi động của chương trình, chỉ đơn giản là 1 MessageBox, ấn ok để tiếp tục



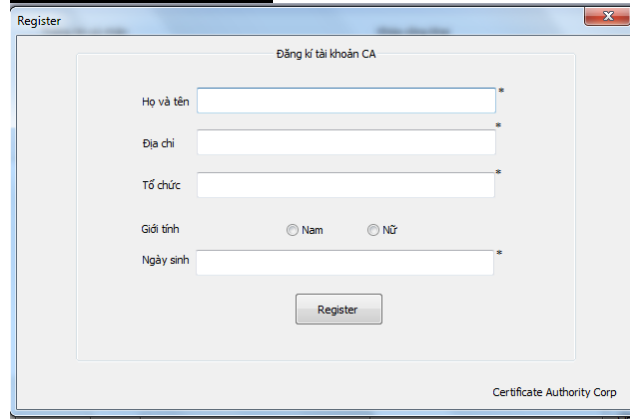
Giao diện người dùng tiến hành mã hóa file

### **Sử dụng:**

#### **Chế độ mã hóa:**

- Bước đầu tiên cần làm là chọn 1 file muốn tiến hành mã hóa với ***Open File..***
- Nhập tên người sử dụng
- Tiến hành mã hóa với Button ***Mã hóa***
- Lưu trữ lại file đã tiến hành mã hóa và file đính kèm( gồm chữ ký và xác thực) với ***Add signature to Document....***
- ***Clear All*** sử dụng để xóa nội dung trên các trường Edit Box.

#### **Chế độ đăng ký:**

The image shows a 'Register' dialog box titled 'Đăng ký tài khoản CA'. It contains several input fields: 'Họ và tên' (Last name and first name), 'Địa chỉ' (Address), 'Tổ chức' (Organization), 'Giới tính' (Gender) with radio buttons for 'Nam' (Male) and 'Nữ' (Female), and 'Ngày sinh' (Date of birth). There is a 'Register' button at the bottom. The dialog box is from 'Certificate Authority Corp'.

- Trên Menu Bar chọn Mode-> Certificate Registration
- Điền đầy đủ thông tin đăng ký và ấn Register để lưu trữ thông tin người dùng

Giao diện gồm các trường Edit Box hiển thị các khóa và chữ ký để người dùng có thể sử dụng những dữ liệu đó vào mục đích khác nếu muốn.

### **3.Hướng phát triển**

- Khắc phục hạn chế xảy ra thuộc phương diện kỹ thuật lập trình, tối ưu hóa phương pháp truy xuất cơ sở dữ liệu.
- Bổ xung tính năng gửi mail cho chương trình sử dụng phương thức SMTP