



TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

Môn: Bảo Mật Thông Tin

Bài thực hành số 3



Bài 1: Hiện thực thuật toán hàm băm MD5 với yêu cầu sau:

1.1 Cho phép người dùng nhập username và password.

1.2 Dùng thuật toán MD5 băm username , password và lưu vào File

1.3 Dùng username và password đăng nhập , chứng thực với File đã ghi username, password.

I. GIỚI THIỆU THUẬT TOÁN MD5:

MD5 (Message - Digest - algorithm 5) giải thuật tiêu hóa tập tin là một chuẩn Internet (RFC 1321). Có khả năng băm mã hóa tập tin bất kỳ thành chuỗi HEX 32 ký tự, tương đương 128-bit (mỗi ký tự hex 4-bit x 32 ký tự = 128 bit).

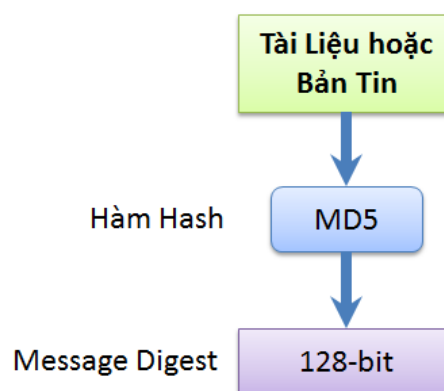
Hoặc có thể định nghĩa theo cách khác. MD5 là cách căn bản để lấy chùm ký tự (là digest, alphabetic hay gì khác), được gọi là string nhập vào và cho ra là 32 ký tự hexa.

(0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f).

MD5 được thiết kế bởi Ronald Rivest vào năm 1991 để thay thế cho hàm băm trước đó, MD4. Vào năm 1996, người ta phát hiện ra một lỗ hổng trong MD5; trong khi vẫn chưa biết nó có phải là lỗi nghiêm trọng hay không, những chuyên gia mã hóa bắt đầu đề nghị sử dụng những giải thuật khác, như SHA-1 (khi đó cũng bị xem là không an toàn). Trong năm 2004, nhiều lỗ hổng hơn bị khám phá khiến cho việc sử dụng giải thuật này cho mục đích bảo mật đang bị đặt nghi vấn.

✚ ĐẶC ĐIỂM MD5

Việc tính MD đơn giản, có khả năng xác định được file có kích thước nhiều Gb.



Không có khả năng tính ngược, khi tìm ra MD.

Do bản chất ngẫu nhiên của hàm băm và số lượng cực lớn các giá trị hash có thể, nên hầu như không có khả năng hai bản tin phân biệt có cùng giá trị hash.

Giá trị MD phụ thuộc vào bản tin tương ứng.

Một chuỗi chỉ có duy nhất một hash.

Giá trị MD phụ thuộc vào tất cả các bit của bản tin tương ứng.

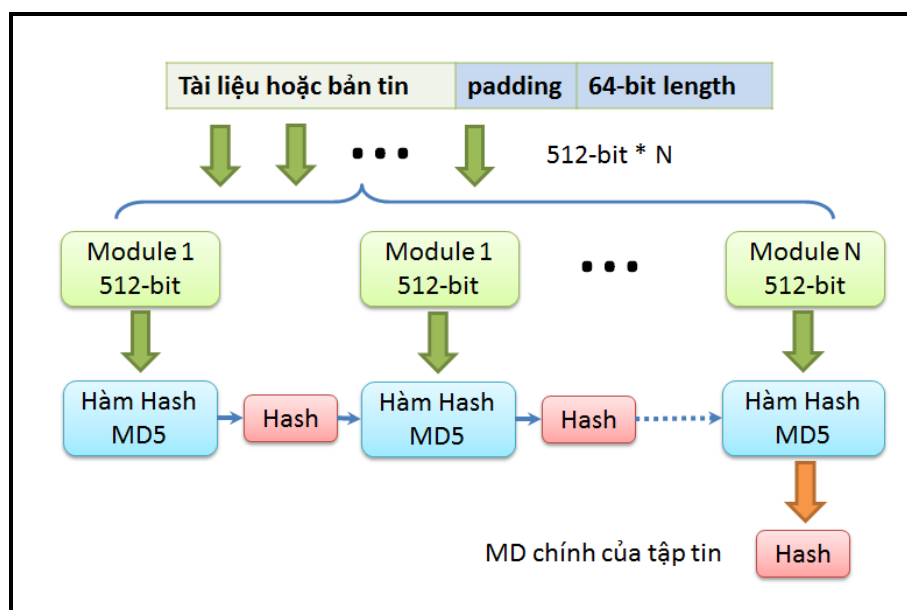
Ví dụ :

love is blue → 03d4ad6e7fee3f54eb46b5ccde58249c

love is Blue → 82b76f8eeb4a91aa640f9a23016c7b1c

II. THUẬT TOÁN

Giải thuật MD5 chính hoạt động trên trạng thái 128-bit, được chia thành 4 từ 32-bit, với ký hiệu A, B, C và D. Chúng được khởi tạo với những hằng số cố định. Giải thuật chính sau đó sẽ xử lý các khối tin 512-bit, mỗi khối xác định một trạng thái. Quá trình xử lý khối tin bao gồm bốn giai đoạn giống nhau, gọi là vòng; mỗi vòng gồm có 16 tác vụ giống nhau dựa trên hàm phi tuyến F, cộng Module, và dịch trái.



Thực hiện qua các bước sau:

Bước 1: Thêm các bit vào chuỗi

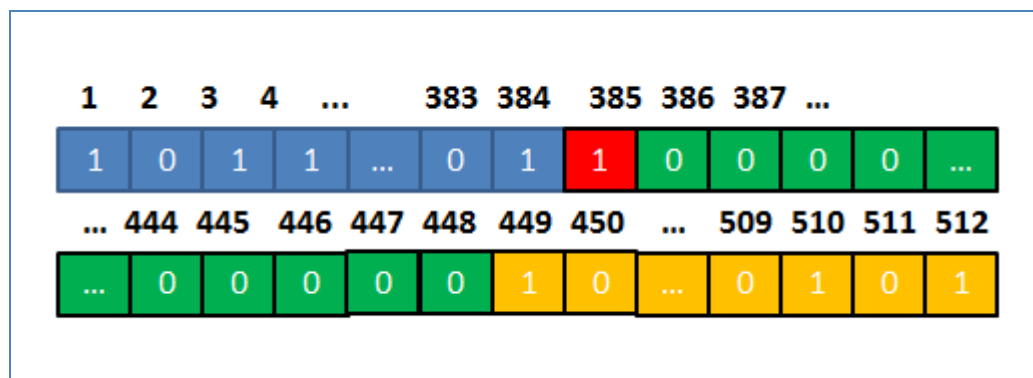
Thực hiện nối dài thông điệp. (theo hình vẽ thông điệp là B) để chi nhỏ thành các module 512.



- ✚ Thêm bit '1' vào cuối thông điệp để đánh dấu.
- ✚ Thêm vào k bit '0' sao cho $(b \text{ bit} + \text{bit } 1 + k \text{ bit } 0) \bmod 512 = 448$
- ✚ 64 bit tiếp theo sẽ được thêm vào biểu thị chiều dài của chuỗi bit ban đầu.

$$(B \text{ bit} + \text{bit '1'} + k \text{ bit '0'} + 64 \text{ bit chiều dài}) \bmod 512 = 0$$

Ví dụ: Ta có chuỗi 384bit



Quá trình thêm bit

Bước 2: Khởi tạo bộ đệm MD

Một bộ đệm 4 word (A,B,C,D) được dùng để tính mã số thông điệp. Ở đây mỗi A,B,C,D là một thanh ghi 32 bit. Những thanh ghi này được khởi tạo theo những giá trị hex sau (các byte thấp trước) :

word A : 01 23 45 67

word B : 89 ab cd ef

word C : fe dc ba 98

word D : 76 54 32 10

Bước 3: Xử lý thông điệp theo từng khối 16 word

Trước hết ta định nghĩa các hàm phụ, các hàm này nhận đầu vào là 3 word 32 bit và tạo ra một word 32 bit.

$$\begin{aligned}F_1(X, Y, Z) &= (X \wedge Y) \vee (\neg X \wedge Z) \\F_2(X, Y, Z) &= (X \wedge Z) \vee (Y \wedge \neg Z) \\F_3(X, Y, Z) &= X \oplus Y \oplus Z \\F_4(X, Y, Z) &= Y \oplus (X \vee \neg Z)\end{aligned}$$

Với $\oplus, \wedge, \vee, \neg$ lần lượt là **XOR, AND, OR, NOT**

Đây là quá trình thực hiện xử lý của 4 hàm F ở trên:

Quá trình này sử dụng một bảng có 64 giá trị $T[1 \dots 64]$ được tạo ra từ hàm sin. Gọi $T[i]$ là phần tử thứ i của bảng, thì $T[i]$ là phần nguyên của $4294967296 * |\sin(i)|$, i được tính theo radian.

Thực hiện:

/ Xử lý mỗi khối 16 word */*

For ($i = 0$ to $N/16-1$) do

/ Copy block i into X. */*

For $j = 0$ to 15 do

Set $X[j]$ to $M[i*16+j]$.

end */* of loop on j */*

/ Lưu A vào AA, B vào BB, C vào CC, D và DD . Làm buffer */*

$AA = A$

$BB = B$

$CC = C$

$DD = D$

Quá trình thực hiện qua các vòng

Vòng 1

- **[abcd k s t]** là bước thực hiện của phép toán

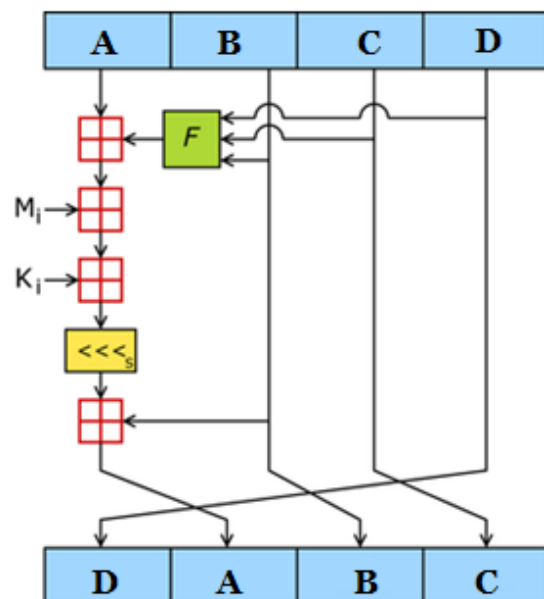
$$a = b + ((a + F(b, c, d) + X[k] + T[i]) \lll s)$$

```
/* Do the following 16 operations. */
[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
```

- Nhận xét: Vòng Một dùng hàm F_1 với giá trị
 - t từ 1..16 và
 - k từ 0..15

Một Thao Tác MD5 - MD5

- Một thao tác MD5—MD5 bao gồm 64 tác vụ thể này, nhóm trong 4 vòng 16 tác vụ. F là một hàm phi tuyến; một hàm được dùng trong mỗi vòng. Mi chỉ ra một khối tin nhập vào 32-bit, và Ki chỉ một hằng số 32-bit, khác nhau cho mỗi tác vụ.
- $\lll s$ chỉ sự xoay bit về bên trái s đơn vị; s thay đổi tùy theo từng tác vụ. \boxplus chỉ cộng thêm với modulo 2^{32} .



Vòng 2

- [abcd k s t] là bước thực hiện của phép toán
$$a = b + ((a + F_2(b, c, d) + X[k] + T[i]) \lll s)$$

```
/* Do the following 16 operations. */
[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]
```

- Nhận xét: Vòng Một dùng hàm F_2 với giá trị
 - t từ 17..32 và
 - $k = (1+5t) \bmod 16$

Vòng 3

- [abcd k s t] là bước thực hiện của phép toán
$$a = b + ((a + F_3(b, c, d) + X[k] + T[i]) \lll s)$$

```
/* Do the following 16 operations. */
[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
```

- Nhận xét: Vòng Một dùng hàm F_2 với giá trị
 - t từ 33..48 và
 - $k = (5+3t) \bmod 16$

Vòng 4

- [abcd k s t] là bước thực hiện của phép toán
$$a = b + ((a + F_4(b, c, d) + X[k] + T[i]) \lll s)$$

```
/* Do the following 16 operations. */
[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]
```

- Nhận xét: Vòng Một dùng hàm F_2 với giá trị
 - t từ 49..64 và
 - $k = 7t \bmod 16$

/* Then perform the following additions. (That is increment each of the four registers by the value it had before this block was started.) */

/* Sau đó làm các phép cộng sau. (Nghĩa là cộng vào mỗi thanh ghi giá trị của nó trước khi vào vòng lặp) */

A = A + AA

B = B + BB

C = C + CC

D = D + DD

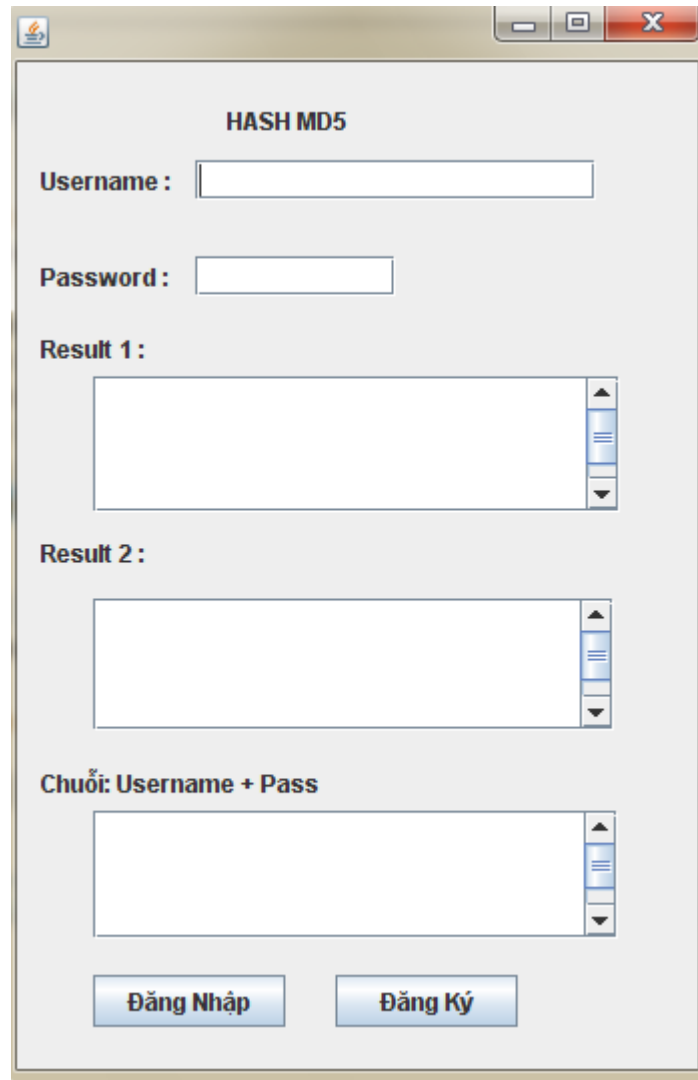
end /* of loop on i */

Bước 4: In ra

Mã số thông điệp được tạo ra là A,B,C,D. Nghĩa là chúng ta bắt đầu từ byte thấp của A, kết thúc với byte cao của D.

III. THUẬT TOÁN

III.1 Thiết Kế Form



The screenshot shows a Java Swing window titled "HASH MD5". Inside the window, there is a form with the following components:

- A label "Username:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A label "Result 1:" followed by a large text area with a vertical scrollbar.
- A label "Result 2:" followed by a large text area with a vertical scrollbar.
- A label "Chuỗi: Username + Pass" followed by a large text area with a vertical scrollbar.
- Two buttons at the bottom: "Đăng Nhập" (Login) and "Đăng Ký" (Register).

III.2 Thư Viện cần sử dụng

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.security.MessageDigest;
import javax.swing.JOptionPane;
```

III.3 Viết hàm xử lý sự kiện

III.3.1 Xử lý sự kiện “Đăng Ký”


```

private void bntDangKyActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        String user = txtuser.getText();
        String pass = txtpass.getText();
        String bam = "";
        bam = user + pass;
        MessageDigest md = MessageDigest.getInstance("MD5");
        md.update(bam.getBytes());
        byte[] byteData = md.digest();
        //convert the byte to hex format method 1
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < byteData.length; i++) {
            sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
        }
        System.out.println("Digest(in hex format):: " + sb.toString());
        txtbam1.setText(sb.toString());
        //convert the byte to hex format method 2
        StringBuffer hexString = new StringBuffer();
        for (int i = 0; i < byteData.length; i++) {
            String hex = Integer.toHexString(0xff & byteData[i]);
            if (hex.length() == 1) {
                hexString.append('0');
            }
            hexString.append(hex);
        }
        System.out.println("Digest(in hex format):: " + hexString.toString());
        txtbam2.setText(hexString.toString());
        txtgoc.setText(bam.toString());

        //Viết chức năng ghi File
        BufferedWriter bw = null;
        //ghi van ban da ma hoa
        String fileName = "D:\\BamMD5.txt";
        //luu van ban
        bw = new BufferedWriter(new FileWriter(fileName));
        // ghi van ban
        bw.write(hexString.toString());
        bw.close();
        JOptionPane.showMessageDialog(null, "Bạn Đã Đăng Ký Thành Công .Vui lòng Đăng nhập lại !!!");

    } catch (Exception ex) {
        System.out.println(" Loi bam username và password : " + ex);
    }
}

```

III.3.2 Xử lý sự kiện đăng nhập

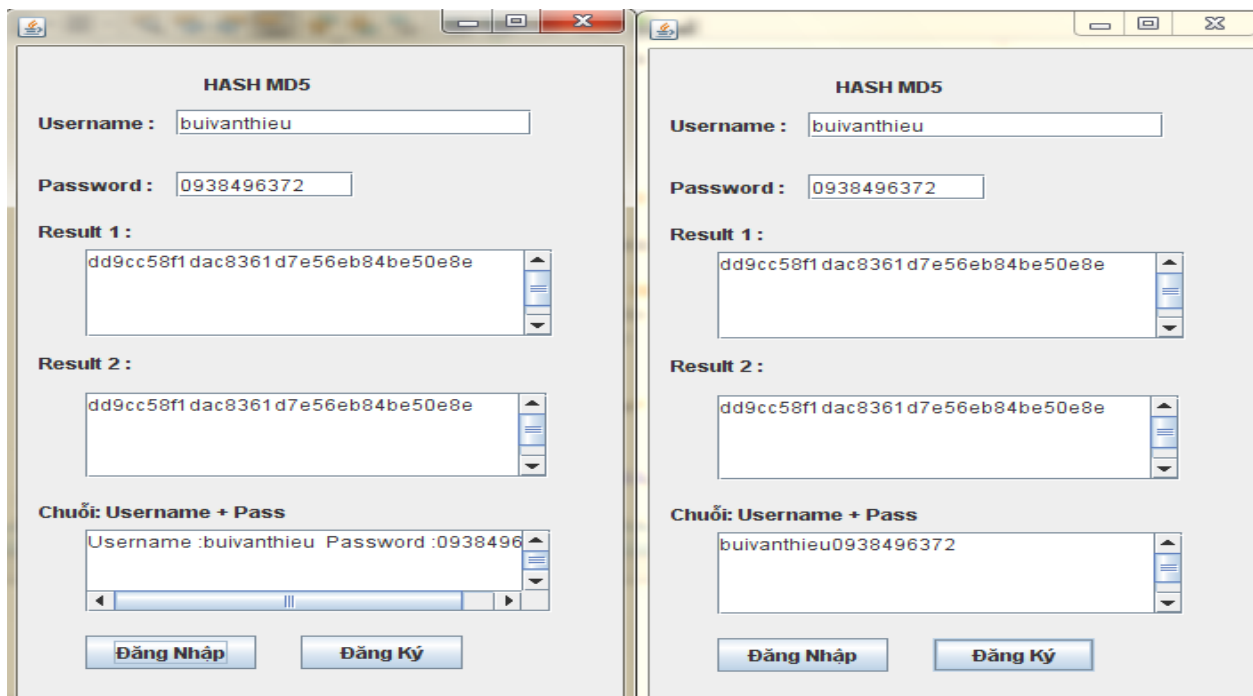
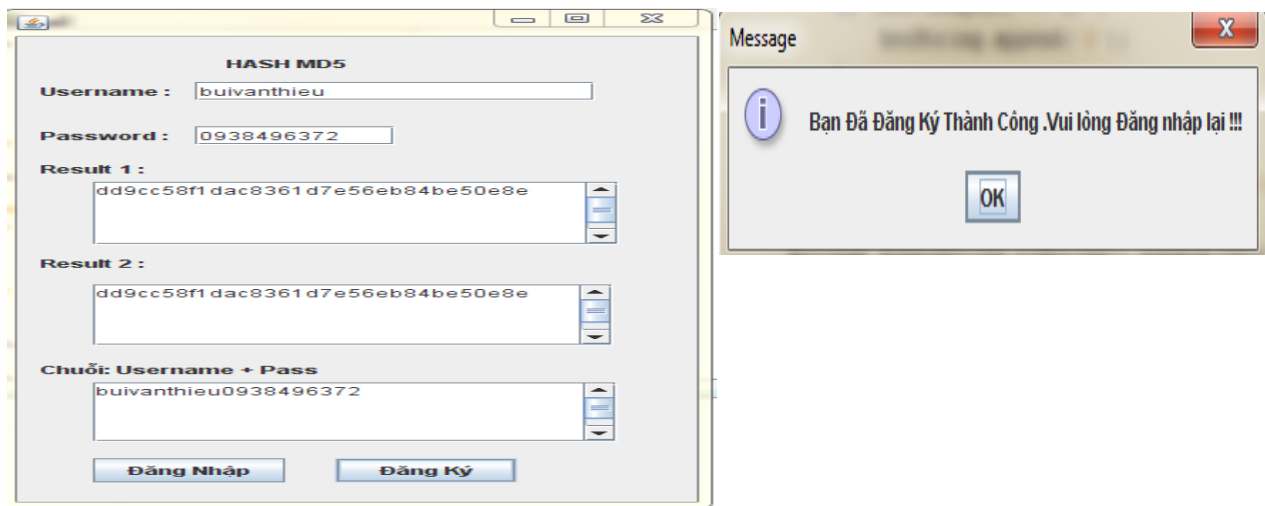
```
private void bntDangNhapActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String user = txtuser.getText();  
    String pass = txtpass.getText();  
    String bam = "";  
    bam = user + pass;  
    //mở File đã lưu username và password  
    BufferedReader br = null;  
    String fileName = "D:\\\\BamMD5.txt"; //GEN-  
    try{  
        br = new BufferedReader( new FileReader(fileName));  
        StringBuffer sb = new StringBuffer();  
        char[] ca = new char[5];  
        while (br.ready()) {  
            int len = br.read(ca);  
            sb.append(ca, 0, len);  
        }  
        br.close();  
        //hiển thị File đã lưu  
        System.out.println("chung thuc :" + " " + sb);  
        String chuoi = sb.toString();  
  
        // Thực hiện băm username và pass cho người dùng đăng nhập  
        MessageDigest md = MessageDigest.getInstance("MD5");  
        md.update(bam.getBytes());  
        byte[] byteData = md.digest();  
        StringBuffer hexString = new StringBuffer();  
        for (int i = 0; i < byteData.length; i++) {  
            String hex = Integer.toHexString(0xff & byteData[i]);  
            if (hex.length() == 1) {  
                hexString.append('0');  
            }  
            hexString.append(hex);  
        }  
        System.out.println("Bam username và password :" + " " + hexString.toString());  
        // Thực hiện so sánh username và password  
        Boolean k=hexString.toString().equals(chuoi);  
        if(k==true)  
        {  
            JOptionPane.showMessageDialog(null, " Đăng Nhập Thành Công");  
            // hiển thị username và password bị băm  
            txtbam1.setText(hexString.toString());  
        }  
    }  
}
```

```

//hiển thị mã băm lưu ở File
txtbam2.setText(chuoi);
txtgoc.setText("Username :"+user+" " + " Password :"+ pass );
}
else
    JOptionPane.showMessageDialog(null, " Đăng Nhập Thất bại");
} catch (Exception ex) {
    System.out.println(" Loi Dang Nhap :"+ ex);
}
}
}

```

IV.Kiểm Tra Kết Quả



Bài 2: Hiện thực thuật toán hàm băm SHA với yêu cầu sau:

Nhập chuỗi và sử dụng thuật than SHA băm chuỗi theo 2 cách.

Hướng dẫn Thuật Toán

Thiết kế Frame:

The image shows a graphical user interface (GUI) for a SHA hashing program. The window has a title bar that reads "Chương Trình Hiện Thực Hàm Băm SHA". Inside the window, there are three main components on the left side, each with a label and a corresponding input/output field:

- Nhập Chuỗi**: A text input field, currently highlighted with a green background.
- Hash SHA C1**: A text area for displaying the first hash result.
- Hash SHA C2**: A text area for displaying the second hash result.

At the bottom of the window, there are two buttons:

- BămSHA**: The button to execute the hashing operation.
- Thoát**: The button to exit the program.

```

private void bntbamSHAActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String chuoi = "";
        chuoi = txtchuoi.getText();
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        md.update(chuoi.getBytes());
        byte byteData[] = md.digest();
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < byteData.length; i++) {
            sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
        }

        System.out.println("Hex format1 : " + sb.toString());
        txtsha1.setText(sb.toString());

        //convert the byte to hex format method 2
        StringBuffer hexString = new StringBuffer();
        for (int i=0;i<byteData.length;i++) {
            String hex=Integer.toHexString(0xff & byteData[i]);
            if(hex.length()==1) hexString.append('0');
            hexString.append(hex);
        }
        System.out.println("Hex format2 : " + hexString.toString());
        txtsha2.setText(hexString.toString());
    } catch (NoSuchAlgorithmException ex) {
        Logger.getLogger(FrameSHA.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

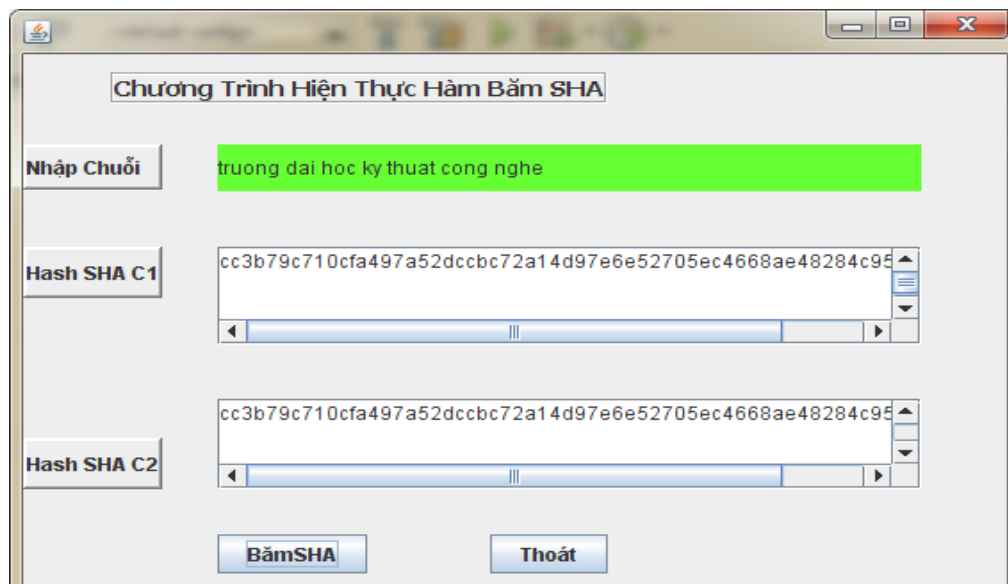
Xử lý sự kiện thoát Form

```

private void bntthoatActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(WIDTH);
}

```

Kết Quả



Bài 3: Hiện thực thuật toán hàm băm SHA với yêu cầu sau:

- 3.1 Cho phép người dùng nhập username và password.**
- 3.2 Dùng thuật toán SHA băm username , password và lưu vào File**
- 3.3 Dùng username và password đăng nhập , chứng thực với File đã ghi username, password. (sử dụng hướng dẫn bài 1 và bài 2...)**