

Part 8 – Cryptography And Efficient Flow Code

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryodesign.asia>

www.suryodesign.wordpress.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Perkembangan Teknologi di Indonesia sangatlah pesat , teknologi menentukan sekali kemajuan di suatu perusahaan, baik di mulai dari proses bisnis dalam perusahaan itu sendiri atau iklan maupun advertising ataupun merupakan webprofile pada suatu halaman internet , dalam prosesnya sendiri lalu lintas informasi dalam dunia maya perlu dijaga , hal ini sangat penting , karena informasi tersebut merupakan informasi yang penting bagi suatu pihak tertentu baik dalam penerimaan maupun pengiriman proses informasi.

Dalam hal ini kita membutuhkan pengamanan , banyak hal yang di lakukan oleh programmer untuk mengamankan data yang ada mulai dari menggunakan SSL , SESSION , COOKIES , ataupun ENCRYPTION dan Kriptografi dan alternative sejenisnya. Dan bagaimana kita menerapkan nya di dalam pemrograman kita , disini saya akan mengajarkannya dan memberitahu bagaimana menerapkan inheritance dan menggunakannya di dalam class kita , sehingga berapa banyak nya form yang kita miliki kita tetap akan menggunakan 1 class untuk show data dan 1 class untuk insert apapun jenisnya ☺.

1

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

harap Sebelum membaca artikel ini anda telah mengerti mengenai konsep manipulation data pada SQL 2005 dan mengerti konsep dasar penggunaan Class dan inheritance yang telah saya jelaskan pada artikel sebelumnya , bagi yang belum mengerti dapat anda download di www.suryodesign.asia/freedownload , harap pelajari konsep2 pada artikel sebelumnya atau anda akan mengalami kesusahan atau bahkan tersesat pada artikel ini karena tidak mengerti konsep dan penggunaan class maupun cara passing parameter .

Isi

Disini saya akan membahas mengenai Kriptografi, Kriptografi disini berasal dari bahasa Yunani yaitu Kryptos yang berarti tersembunyi dan grafo yang berarti menulis, dan kemudian Kriptografi sendiri artinya merupakan teknik untuk menyamarkan atau membuat informasi / pesan yang telah telah diinput seolah olah tidak terlihat oleh pihak lain. Sekalipun mereka bisa mendapatkan informasi yang ada tetapi mereka tidak akan mendapatkan isi pesan yang asli dari message tersebut.

Kita menggunakan MD5 yang merupakan singkatan dari Message Digest 5 (MD-5) , disini nomor "5" merupakan versi perbaikan dari versi2 yang ada pada sebelumnya , MD5 di buat oleh Ron Rivest pada tahun 1991 , untuk menggantikan hash function sebelumnya yaitu MD4 yang dibuat pada tahun 1991 dan kemudian pada tahun 1996 kemudian ditemukan MD5 , dan kemudian pada tahun 2007 dilakukan penelitian oleh Arjen Lestra untuk membuat SSL Sertificate validate untuk menambah atau mensupport keamanan pada MD5 ini .

Apa itu MD5 ?

MD5 adalah teknik / formula yang mengubah kalimat yang ada dan kemudian di hashing menjadi satu arah yang menghasilkan hash 128 bit "finger print" atau "message digest" yang maksudnya secara langsung di konversi dan tidak meninggalkan jejak sehingga tidak dapat di konversi kembali ke dalam bentuk aslinya. MD5 merupakan salah satu cara untuk mengubah data integrity yang berguna untuk menyimpan password atau data yang sensitive atau hanya boleh digunakan oleh beberapa pihak saja.

Apakah MD5 merupakan enkripsi ?

Ada beberapa yg bilang MD5 merupakan teknik Enkripsi, tetapi ada yg bilang jg bukan , hal ini merupakan suatu perbedaan pendapat dimana satu pihak mengatakan bahwa yang namanya enkripsi mesti dapat di dekripsi untuk mendapatkan value / nilai aslinya , dan ada yg mengatakan ini hanyalah jalan satu arah untuk mengacak atau mengubah kalimat yang ada dan tidak meninggalkan kalimat aslinya dan kita tidak dapat menggunakan md5sum untuk mendapat kembali kalimat aslinya.

Okay , sekarang kita memiliki kalimat seperti password yang telah di konversi, dapatkan kita mendapatkannya isi kalimat original tersebut dengan cara seperti Brute Force ?

seperti system password yang ada , kita dapat melakukan brute force yang ada , tetapi bagaimanapun MD5 sum merupakan 128 bit space yang artinya kita membutuhkan 2^{128} kemungkinan untuk mendapatkan aslinya, bias dibutuhkan waktu yang lama sekali untuk kalimat yang diinput merupakan gabungan number dan alphabet yang dibedakan huruf besar atau kecilna dan makin panjang kalimatnya pun mempengaruhi lamanya hasil pencarian.

Okay. Tetapi bagaimana bila ada MD5 dictionary yang dibuat oleh seseorang untuk memecahkan MD5 yang kita miliki , dan mengetahui password MD5 aslinya ?

ini memungkinkan saja , tetapi ini membutuhkan waktu yang lama juga karena dictionary ini sendiri terdiri dari alphet (huruf kecil dan besar) dan juga number yang berarti ada kemungkinan 46,656,000,000 entries data dan semua terdiri dari 32 characters setiap kalimat entriesnya (jangan lupa karena MD5 memiliki panjang 32 karakter untuk hasil output nya). Ini membutuhkan sekitar 1 Terabyte data untuk menyimpan dan entah dibutuhkan waktu berapa lama untuk mencarinya.

Berikut adalah step by step MD5 Alghotritm yang saya dapat dari narasumber pada salah satu situs yang membahas tentang Cryptography , sumber ini saya biarkan dalam bentuk original agar tidak terjadi kesalahan dalam translate nya .

MD5 Algorithm Overview

MD5 algorithm is well described in RFC 1321 - The MD5 Message-Digest Algorithm, see <http://www.ietf.org/rfc/rfc1321.txt>. Below is a quick overview of the algorithm.

MD5 algorithm consists of 5 steps:

Step 1. Appending Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

- The original message is always padded with one bit "1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

Step 2. Appending Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

- The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
- Break the 64-bit length into 2 words (32 bits each).

- The low-order word is appended first and followed by the high-order word.

Step 3. Initializing MD Buffer. MD5 algorithm requires a 128-bit buffer with a specific initial value. The rules of initializing buffer are:

- The buffer is divided into 4 words (32 bits each), named as A, B, C, and D.
- Word A is initialized to: 0x67452301.
- Word B is initialized to: 0xEFCDAB89.
- Word C is initialized to: 0x98BADCFE.
- Word D is initialized to: 0x10325476.

Step 4. Processing Message in 512-bit Blocks. This is the main step of MD 5 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, 4 rounds of operations are performed with 16 operations in each round. This step can be described in the following pseudo code slightly modified from the RFC 1321's version:

Input and predefined functions:

A, B, C, D: initialized buffer words

$F(X,Y,Z) = (X \text{ AND } Y) \text{ OR } (\text{NOT } X \text{ AND } Z)$

$G(X,Y,Z) = (X \text{ AND } Z) \text{ OR } (Y \text{ AND } \text{NOT } Z)$

$H(X,Y,Z) = X \text{ XOR } Y \text{ XOR } Z$

$I(X,Y,Z) = Y \text{ XOR } (X \text{ OR } \text{NOT } Z)$

$T[1, 2, \dots, 64]$: Array of special constants (32-bit integers) as:

$T[i] = \text{int}(\text{abs}(\sin(i)) * 2^{32})$

$M[1, 2, \dots, N]$: Blocks of the padded and appended message

$R1(a,b,c,d,X,s,i)$: Round 1 operation defined as:

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

$R2(a,b,c,d,X,s,i)$: Round 1 operation defined as:

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

$R3(a,b,c,d,X,s,i)$: Round 1 operation defined as:

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

$R4(a,b,c,d,X,s,i)$: Round 1 operation defined as:

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

Algorithm:

For $k = 1$ to N do the following

AA = A

BB = B

CC = C

DD = D

$(X[0], X[1], \dots, X[15]) = M[k]$ /* Divide $M[k]$ into 16 words */

/* Round 1. Do 16 operations. */

$R1(A,B,C,D,X[0], 7, 1)$

$R1(D,A,B,C,X[1], 12, 2)$

$R1(C,D,A,B,X[2], 17, 3)$

```
R1(B,C,D,A,X[ 3],22, 4)
R1(A,B,C,D,X[ 4], 7, 5)
R1(D,A,B,C,X[ 5],12, 6)
R1(C,D,A,B,X[ 6],17, 7)
R1(B,C,D,A,X[ 7],22, 8)
R1(A,B,C,D,X[ 8], 7, 9)
R1(D,A,B,C,X[ 9],12,10)
R1(C,D,A,B,X[10],17,11)
R1(B,C,D,A,X[11],22,12)
R1(A,B,C,D,X[12], 7,13)
R1(D,A,B,C,X[13],12,14)
R1(C,D,A,B,X[14],17,15)
R1(B,C,D,A,X[15],22,16)

/* Round 2. Do 16 operations. */
R2(A,B,C,D,X[ 1], 5,17)
R2(D,A,B,C,X[ 6], 9,18)
R2(C,D,A,B,X[11],14,19)
R2(B,C,D,A,X[ 0],20,20)
R2(A,B,C,D,X[ 5], 5,21)
R2(D,A,B,C,X[10], 9,22)
R2(C,D,A,B,X[15],14,23)
R2(B,C,D,A,X[ 4],20,24)
R2(A,B,C,D,X[ 9], 5,25)
R2(D,A,B,C,X[14], 9,26)
R2(C,D,A,B,X[ 3],14,27)
R2(B,C,D,A,X[ 8],20,28)
R2(A,B,C,D,X[13], 5,29)
R2(D,A,B,C,X[ 2], 9,30)
R2(C,D,A,B,X[ 7],14,31)
R2(B,C,D,A,X[12],20,32)

/* Round 3. Do 16 operations. */
R3(A,B,C,D,X[ 5], 4,33)
R3(D,A,B,C,X[ 8],11,34)
R3(C,D,A,B,X[11],16,35)
R3(B,C,D,A,X[14],23,36)
R3(A,B,C,D,X[ 1], 4,37)
R3(D,A,B,C,X[ 4],11,38)
R3(C,D,A,B,X[ 7],16,39)
R3(B,C,D,A,X[10],23,40)
R3(A,B,C,D,X[13], 4,41)
R3(D,A,B,C,X[ 0],11,42)
R3(C,D,A,B,X[ 3],16,43)
R3(B,C,D,A,X[ 6],23,44)
R3(A,B,C,D,X[ 9], 4,45)
R3(D,A,B,C,X[12],11,46)
R3(C,D,A,B,X[15],16,47)
R3(B,C,D,A,X[ 2],23,48)

/* Round 4. Do 16 operations. */
R4(A,B,C,D,X[ 0], 6,49)
R4(D,A,B,C,X[ 7],10,50)
R4(C,D,A,B,X[14],15,51)
R4(B,C,D,A,X[ 5],21,52)
R4(A,B,C,D,X[12], 6,53)
```

```
R4(D,A,B,C,X[ 3],10,54)
R4(C,D,A,B,X[10],15,55)
R4(B,C,D,A,X[ 1],21,56)
R4(A,B,C,D,X[ 8], 6,57)
R4(D,A,B,C,X[15],10,58)
R4(C,D,A,B,X[ 6],15,59)
R4(B,C,D,A,X[13],21,60)
R4(A,B,C,D,X[ 4], 6,61)
R4(D,A,B,C,X[11],10,62)
R4(C,D,A,B,X[ 2],15,63)
R4(B,C,D,A,X[ 9],21,64)
```


```
A = A + AA
B = B + BB
C = C + CC
D = D + DD
End of for loop
```

Output:

A, B, C, D: Message digest

Step 5. Output. The contents in buffer words A, B, C, D are returned in sequence with low-order byte first.

Sekarang langsung saja masuk ke penerapan Cryptography tersebut ke dalam aplikasi. Pertama buat database MsUser pada SQL2005 dengan struktur berikut :

SURYODESIGN-PC\...05 - dbo.MsUser			
Column Name		Data Type	Allow Nulls
	Username	varchar(20)	<input type="checkbox"/>
	PassMD5	varchar(35)	<input checked="" type="checkbox"/>
	PassSHA1	varchar(35)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Setelah anda membuatnya maka kemudian design form input user dan login seperti berikut , aplikasi ini merupakan aplikasi lanjutan dari pertemuan 7 , anda dapat download di web saya bila belum memilikinya di www.suryodesign.asia .

Design Form Input User seperti berikut :

Username	PassMD5	PassSHA1
adli	58c1a701fa3ac19ede3b2699b1a73a7e	/yWAS62pzUAYF/JoHp/Fd13mpGk=
mizone	b2348f469a472692555470f6ade2a2ef	HV8q+0pg0x3Kuhcf/DiYBIDRmpk=
suryo	98d006db0378032df287e4bb6913a13c	wRLAldCoyqRUywuSC3CUB+JNyY=

Username

Password 1

Password 2

Kemudian design Form MenuLogin seperti berikut :

Menu Login C#

Username

Password

Encryption



Kemudian Pada Class2005.Cs tambahkan class induk 2005.cs yang akan kita inherits atau turunkan pada class anaknya , jangan lupa mengenai konsep setter dan getter yang berguna untuk menangkap nilai dan men set nilai , dan jgn lupa jg mengenai konsep static , nonstatic , serta penggunaan public atau private.

```
public class ClassInduk2005
{
    public string prefixNama = "suryo." ;
    public static int _flagLogin = 0;
    public static int flagLogin
    {
        get { return _flagLogin; }
        set { _flagLogin = value; }
    }

    public static string _Nama;
    public static string Nama
    {
        get { return _Nama; }
        set { _Nama = value; }
    }
}
```

Kemudian turunkan pada Class2005 kita dengan menambahkan titik dua (:) atau colon seperti berikut pada Class2005 yang kita miliki pada awalnya :

```
public class Class2005 : ClassInduk2005
```

Setelah ini dilakukan maka kita dapat memanggil baseclass dimana tempat variable di inheritkan / diturunkan kepada form anak.

Kemudian Deklarasi variable yang kita gunakan untuk menampilkan serta insert dan menampung data pada Class2005.cs

```
private static SqlConnection con = new SqlConnection(@"Data
Source=SURYODESIGN-PC\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True");
private static SqlCommand com = new SqlCommand();
private static SqlDataAdapter sda = new SqlDataAdapter();
private static SqlDataReader sdr;
private static DataSet _ds ;

public static DataSet ds
{
    get { return _ds; }
    set { _ds = value; }
}
```



```
private static string _query;
public static string query
{
    get { return _query; }
    set { _query = value; }
}

private static string _Message;
public static string Message
{
    get { return _Message; }
    set { _Message = value; }
}
```

Setelah mendeklarasikan variable diatas sekarang saatnya kita membuat setter dan getter untuk enkripsi MD5 dan SHA1 yang telah diinput,dan kita tidak langsung mengacak input yang ada , tetapi kita menambahkan sedikit ramuan dari input user digabung dengan “suryo.”, kita menambahkan ini sebagai KeyReference agar hasil enkripsi kita tidak sepenuhnya merupakan inputan user asli , tetapi sebenarnya ada KeyReference yang disisipkan disana, anda dapat lebih creative dengan cara anda sendiri , berikut contoh enkripsi MD5:

```
private static string _stringMD5 ;
public static string stringMD5
{
    get
    {
        //String kosong yang digunakan menampung enkripsi
        string ret = String.Empty;
        //deklarasikan cryptography
        MD5CryptoServiceProvider md5Hasher = new MD5CryptoServiceProvider();
        //dapatkan byte data
        byte[] data = System.Text.Encoding.ASCII.GetBytes("suryo." +
            _stringMD5);
        //Encrypt
        data = md5Hasher.ComputeHash(data);
        //convert data dari byte menjadi hex
        for (int i = 0; i < data.Length; i++)
        {
            ret += data[i].ToString("x2").ToLower();
        }
        //Return encoded
    }
}
```

```
return ret;
}
set { _stringMD5 = value ; }
}
```

Berikut Enkripsi untuk SHA1 :

```
private static string _stringSHA1;
public static string stringSHA1
{
get{
string rethash = "";
System.Security.Cryptography.SHA1 hash
= System.Security.Cryptography.SHA1.Create();
System.Text.ASCIIEncoding encoder = new System.Text.ASCIIEncoding();
byte[] combined = encoder.GetBytes("suryo." + _stringSHA1);
hash.ComputeHash(combined);
rethash = Convert.ToBase64String(hash.Hash);
return rethash;
}
}
set { _stringSHA1 = value; }
}
```

Setelah itu selesai , maka untuk aplikasi kita , kita akan menggunakan 1 class saja untuk semua data yang tampil dan semua proses insert yang ada seperti berikut :
Untuk Show Data :

```
public static void tampilData(ref string namaTable)
{
com.Connection = con;
com.CommandType = CommandType.Text;
com.CommandText = _query;
_ds = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(com);
sda.Fill(_ds, namaTable);
}
```

Setelah kita deklarasikan ini , di aplikasi kita untuk menampilkan cukup dengan

```
dataGridBarang.DataSource = Class2005.ds;          dataGridBarang.DataMember = a;
```

Kita tidak perlu Return Dataset dan sejenisnya karena kita dapat langsung mengaksesnya melewati class2005 . banyak cara / modifikasi dari source yang dapat dilakukan , ini tergantung dari programmer itu sendiri lebih menyukai cara yang mana.

Berikut Coding untuk Insert Data :

```
public static void insertData(ref string perintahQuery )
{
    con.Open();
    com.Connection = con;
    com.CommandType = CommandType.Text;
    com.CommandText = perintahQuery;
    com.ExecuteNonQuery();
    con.Close();
}
```

Dan Untuk Memanggilnya dari form insert data, ada beberapa hal yg perlu kita lakukan yaitu definisikan class , konversi password ke dalam md5 dan SHA1 kemudian tentukan table untuk dataset dan baru kemudian insert dan refresh table yang ada seperti berikut :

```
ClassInduk2005 ClassAnak2005 = new ClassInduk2005();
string enkripMd5 = Class2005.stringMD5 = txtPass1.Text;
string enkripSha1 = Class2005.stringSHA1 = txtPass2.Text;
string query = "insert into MsUser values('" + txtUsername.Text + "','" + Class2005.stringMD5 + "','" + Class2005.stringSHA1 + "')";

Class2005.insertData(ref query);
Class2005.query = "select * from msuser";
string a = "user";
Class2005.tampilData(ref a);
dataGridView1.DataSource = Class2005.ds;
dataGridView1.DataMember = a;
```

Setelah Insert Data Sukses saat nya kita membahas mengenai Login, berikut isi classlogin pada class2005.cs pada aplikasi :

```
public static void loginData(ref string txtUsername,ref string txtPass, ref int indexNo)
{
    stringMD5 = txtPass;
    stringSHA1 = txtPass;
```

```
if (con.State != ConnectionState.Open)
{
    con.Open();
}
com.Connection = con;
com.CommandType = CommandType.Text;
if (indexNo == 0)
{
    com.CommandText = "select * from msuser where Username = " + txtUsername
    + " and passMD5 =" + stringMD5 + "";
}
else
{
    com.CommandText = "select * from msuser where Username = " + txtUsername
    + " and PassSHA1 =" + stringSHA1 + "";
}
SqlDataReader sdr = com.ExecuteReader();
if (sdr.HasRows)
{
    Class2005.flagLogin = 1;
}
else
{
    Class2005.flagLogin = 0;
}
if (con.State == ConnectionState.Open)
{
    con.Close();
}
}
```

Kemudian Cara Kita Login dari Form Login aplikasi kita seperti berikut :

```
string txtPass = txtPassLogin.Text;
int indexNo = cmbEnkripsi.SelectedIndex ;
string userLogin = txtUserLogin.Text;
Class2005.loginData(ref userLogin, ref txtPass, ref indexNo);
if (Class2005.flagLogin > 0 )
{
    Form1 form1 = new Form1();
    form1.Show();
    this.Hide();
}
}
```

Penutup

Dibutuhkan banyak latihan dan variasi penulisan yang berbeda untuk mencapai dan mengasa Hinga mencapai tahap piker yang lebih baik dan logika yang matang untuk dapat menciptakan sesuatu , karena itu kita mesti lebih creative dan tidak hanya bergantung pada contoh yang ada pada saat ini , teruslah menuntut ilmu dari banyak literature yang ada dan banyak-banyaklah sharing dengan teman di sekitar anda untuk mendapatkan pengalaman yang lebih baik.

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com
- <http://social.msdn.microsoft.com/>
- www.vb.netforum.com
- <http://www.c-sharpcorner.com>
- <http://www.dotnet-spider.com>
- <http://www.cryptography.org>
- http://en.wikipedia.org/wiki/Hash_function
- <http://en.wikipedia.org/wiki/Md5>
- <http://forums.devnetwork.net>
- http://www.herongyang.com/crypto/message_digest_md5_3.html

Biografi Penulis



M.Suryo Pranoto – Alumni Mahasiswa Perguruan Tinggi Universitas Bina Nusantara , Aktif dalam beberapa komunitas komputer , dan beberapa project terutama berbasis aplikasi seperti VB.Net atau C#,

dan sedang berusaha keras untuk menabung dan berencana untuk melanjutkan Cisco CCNP setelah menyelesaikan CCNA , memiliki hobby untuk sharing mengenai komputer mulai dari software hingga jual beli hardware maupun modding & overclocking computer.

Ym : suryolovetyka

Email : suryodesign@yahoo.co.id

Website : <http://www.suryodesign.asia>

Blog : www.suryodesign.wordpress.com