Cryptography : A Basic Introduction of Hashing and HashAlgo class for beginners – Part 2

Posted on October 25, 2012

4 Votes

In the previous part of this blog we have learned about basic concept of Hashing with its practical example, we've also read about confusing question for beginner which Is both hashing and Encryption are Same ? then after we covered about algorithm classes of Hashing.

Hash Algorithm Classes

Lets have a look at given table for various Hash Algorithm Classes.

Classes	Description
MD5	1. Abstract Class extended by a single Concrete class 2. Used to implement Message Digest hashing algorithm Version 5 3. Support 128-bit hash values 4. Widely deployed and commonly used to verify file Integrity 5. Created by Ron Rivest in 1991 to replace MD4
SHA1	1. Abstract Class extended by a single Concrete class 2. Used to implement SHA1 hashing algorithm 3. created to replace MD5 due to flaws 4. created in 1995 by NSA as U.S Government Standard
KeyedHashAlgorithm	1. Abstract Class extended by a several Concrete class 2. Used to implement the Keyed-Hash Message Authentication Code 3. Created by Mihir Bellare, Ran Canetti and Hugo Krawczyk in 1996 4. Uses an existing hashing algorithm such as SA1 and then iterates twice to doubly hash the output value.
RIPEMD160	1. Abstract Class extended by a single Concrete class 2. Used to implement RACE Integrity Primitive Evaluation Message Digest hashing algorithm 3. Created in Europe by Hans Dobbertin, Antoon Bosselaers and Bart Preneel in 1996 4. Imroved Version of MD5 5. Similar in performance to SHA1
SHA256	1. Abstract Class extended by a single Concrete class 2. Used to implement SHA-256 hashing algorithm 3. Part of SHA hashing algorithm version 2 variants 4. Implement a 256-bit hashing key

SHA384	1. Abstract Class extended by a single Concrete class 2. Used to implement the SHA-384 hashing algorithm 3. Part if SHA hashing algorithm version 2 variants 4. Implements a 384-bit hashing key
SHA512	1. Abstract Class extended by a single Concrete class 2. Used to implement the SHA-512 hashing algorithm 3. Part of SHA hashing algorithm version 2 variants 4. Implements a 512-bit hashing key

Table Content-Reference : *MS Workshop Advanced Foundation of Microsoft .Net Development.(Book)*

As we alrady discus about Hashing by using MD5 class in previous article so here we will go through SHA1 class.

Hashing By Using SHA1 Class

just due to the flaws found in the MD5 hashing algorithm, the SHA1 is considered a better hashing algorithm choice, and the successor of MD5, SHA1 proved that it offers better performance and a more secure hashing algorithm than MD5.

What is SHA1 Class

The SHA1 is one of the Commonly used hasing algorithm which provide Hashing functionality by using SHA1 classes and extending HashAlgorithm classes. The .Net Framework SHA1 hashing algorihm class uses a 160-bit hash key is extended by the SHA1CryptoServiceProvider(CSP) class and the SHA1Managed class.

Note : The SHA1Managed class is a CLR managed based clas which extends the SHA1 class. it provides the same functionality as the SHACryptoServuceProvider class but being a CLR managed class.
Benefit : It performs better, protects data stored in memory and its nicely cleaned up by the garbage collector.

Implementation of SHA1 class

Here given code show you how to use SHA1CryptoserviceProvider for hashing your data. GetSHA1CSPHash is a function which need values as parameter. In function SHA1provider is an instance of SHA1CryptoserviceProvider Class. after creating instance Convert the input string to a byte array and compute the hash, at last managing Loop through each byte of the hashed data and format each one as a hexadecimal string

Note : To access SHA1CryptoserviceProvider class you must need to mention System.Security.Cryptography Namespace.

```
using System.Security.Cryptography;
```

Now create a function

```
public static string GetSHA1CSPHash(string plaintext)
        {
            SHA1CryptoServiceProvider SHA1provider = new SHA1CryptoServiceProvider();
            byte[] hasedvalue = SHA1provider.ComputeHash(Encoding.Default.GetBytes(plaintext));
            StringBuilder str = new StringBuilder();
            for (int counter = 0; counter < hasedvalue.Length; counter++)
            {
                str.Append(hasedvalue[counter].ToString("X1"));
            }
            return str.ToString();
        }
```

In Same way you can create function for Hashing by SHA1Managed Class. you just need to use SHA1Managed class instead of SHA1CryptoServiceProvider Class. You can create Instance like

```
SHA1Managed SHA1provider = new SHA1Managed();
```

How to use Function

you can simply use function by just passing value which need to be hashed as given sample of code.

What Output comes

```
Enter Value for Hashed
Learning SHA1 Hash
Hashed Value : 7D6585B35177A53F95B7BB16D8ACC4EB6F954741
```

How to Verify Hash

```
 static bool VerifySHA1CSPhash(string PlainText, string prevhashedvalue)
        {

            string hashedvalue2 = GetSHA1CSPHash(PlainText);

            // Create a StringComparer an compare the hashes.
            StringComparer strcomparer = StringComparer.OrdinalIgnoreCase;

            if (strcomparer.Compare(hashedvalue2, prevhashedvalue).Equals(0))
            {
                return true;
            }
            else
            {
```

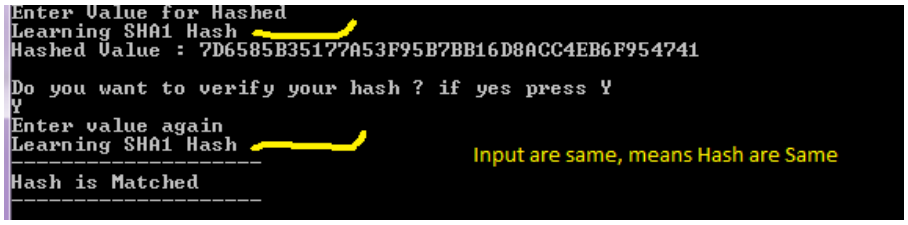
```
                return false;
            }
        }
```

Note : In same manner we can create function to verify Hash which using SHA1Managed class. We just need to corresponding function responsible to hashed data.

we can use given codes to learn how to use both function to check hashed function is same or not.

```
static void Main(string[] args)
      {
          Console.WriteLine("Enter Value for Hashed");
          string yourvalue = Console.ReadLine();

          string strhashed = GetSHA1CSPHash(yourvalue);
          Console.WriteLine("Hashed Value : " + strhashed);

          Console.WriteLine("\nDo you want to verify your hash ? if yes press Y");
          char ch = Convert.ToChar(Console.ReadLine());
          if (ch == 'Y' || ch == 'y')
          {
              Console.WriteLine("Enter value again ");
              string yourvalue2 = Console.ReadLine();

              bool res = VerifySHA1CSPhash(yourvalue2, strhashed);
              Console.WriteLine("---------------------");
              if (res)
              {

                  Console.WriteLine("Hash is Matched");

              }
              else
              {
                  Console.WriteLine("Hash is not Matching ");
              }
              Console.WriteLine("---------------------");
          }
          else { Environment.Exit(1); }

          Console.ReadLine();

      }
```

What output comes



Download SourceCode

Want to download source code : Click Me!

Further Reading

Hashing Function-[Wikipedia]
An Illustrated Guide to Cryptographic Hashes
Basics of Encryption and Hashing-[MSDN]
SHA1Class-[MSDN]

Coming Next

In the next article, I will hopefully come back with more basic details and implementation of HMAC and other Hash Algo classes for beginners. Till then Happy Coding.. 😃

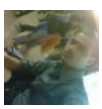
Share this:

- Twitter
- Facebook
- Google
- Tumblr
- Reddit
- Email
- Print
-

Like this:

Like Loading...

Related



About Ravi Ranjan Kumar
An Indian who Living, Loving & Learning Technology with different tastes and willing to share knowledge and thoughts.
View all posts by Ravi Ranjan Kumar →

This entry was posted in .Net, C#, CodeProject, Cryptography and tagged .Net, C#, CodeProject, Cryptography. Bookmark the permalink.

RranjanK's Blogs

The Twenty Ten Theme Blog at WordPress.com.