RranjanK's Blogs

Life between Code and Coffee

Cryptography : A Basic Introduction of Hashing and HashAlgo class for beginners – Part 1

Posted on October 20, 2012

4 Votes

Hashing

Hash is a kind of process, signature, function which is responsible for translating information into a cryptic value. The concept of hash and encryption is almost same. In practical view Hash is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string. Hashing is also known for its unidirectional process because it is not require dehashing or decrypting to get back data. In hashing the data which is needed to be encoded is often called the “message,” and the outcome of hash value after processing is sometimes called the message digest or simply digests.
While hashing message, an algorithm is utilize which work is to map input values to a series of known output values. so given then same series of input values, a hash algorithm always produces the same output values. Hashing is an industry supported standard similar to encryption.

Plain Text
Hash Function

Fig : Hashing Concept

Animation Pic – Hash Function

While creating and designing hash functions, we generally came across with ways:

- First one is we have nothing idea and info about the distribution of the incoming keys.
  In such situation, hash function proved its importance because its evenly distributes the key range across the hash table.
- Second one is we have little bit idea and info about the distribution of the incoming keys.
  In such situation, we should use a distribution-dependent hash function that can avoids assigning clusters of

related key values to the same hash table slot.

Practical working of Hashing

As we know Hashing takes any amount of values(Plain Text and Binary) and then creates a constant-length hash representing a checksum for the data so in sense hashing something is a way of turning something (usually a key or a password) into a (usually fixed length string of characters.

Let's take an example which explains the working process of hashing.
We all daily log on our system either windows/linux/Mac/etc we have to authenticate our self in case if password is set for login. So what we think our password is kept private and secure so no one can trace or stole it. Lots of question in mind like

How we get authenticated by entering password ?

To know the solution of above question we should know the working process of hashing. While creating user account the password which is given by user is not stored by System. Actually a hash value of password is stored. Its all gone through consistent Hashing algorithm to hash plain text to hashed value. So whenever a user enters a password for authentication purpose, this password is not transported or stored. A hash of the password is transported for authentication instead of transporting the password in clear text. System Hashes the password entered by user by using the same hashing algorithm that was used while creating password when user create account.
It is clear System compares the entered hash value while login to the hash value stored while creating account. If the hashes matched the user is authenticated otherwise not authenticated.

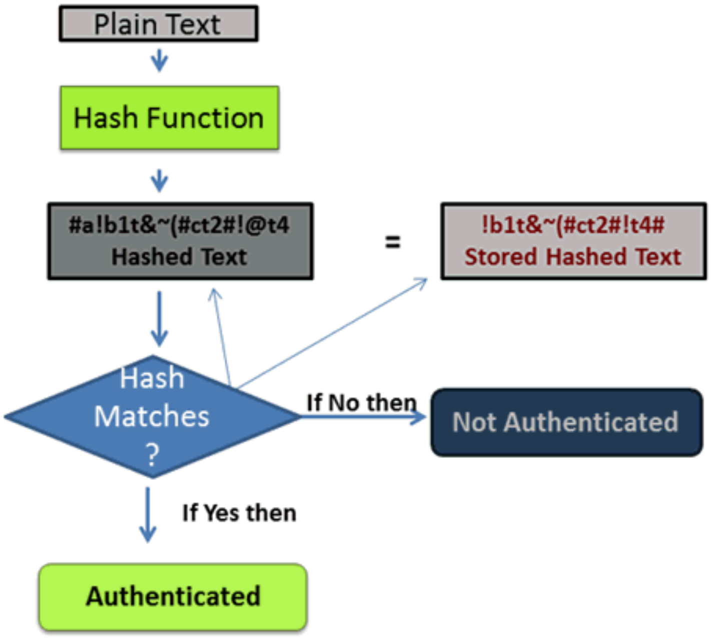


Fig : Flow Chart Example of Hash work

Animated Pic – Flow Chart Example of Hash work

Should we want to hash everything?

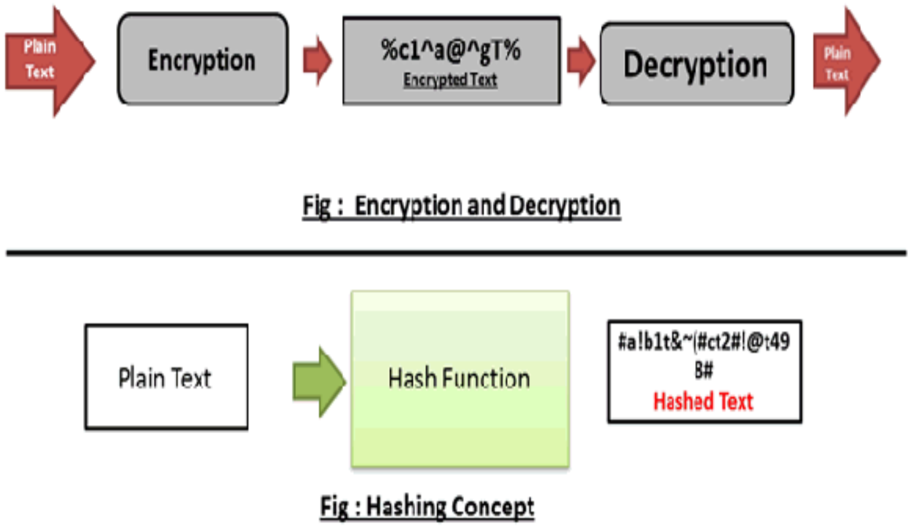
No doubt, it's a common assumption to use Hash to secure everything because of its powerful algorithm and way of hashing data's to fixed constant length.
Why not we can but to make hashing more secure, we should always use salt values. We must also store salt value and keep it confidential and secure. So in such case whenever a hashed value needs to be verified, the input value is combined with the salt value and hashed to generate an output value. So then output hashed value is compared to stored hash value. In sense we can hash a our plain text as long as salt is correctly.
For more info, Must read an awesome article
dont hash code

Is both hashing and Encryption are Same ?

Is hashing and Encryption both are Same ?
Hmm..
I think No..
Lets see how its different 😊

Note – I used Google to get this cartoon pic.

Its usually a common reaction or can be a question for beginner who just put their step in Cryptography. Whatever knowledge of hashing is vital for developers to gain a complete understanding of security and cryptography and being beginner we should have almost basic idea about what hash called and how is different from Encryption.
As we read about Encryption in previous article series-Cryptography : Symmetric Encryption by Symmetric Algorithm Classes–Part 1 is that Encryption is a scheme where an intelligible text is made unintelligible using a secure key. The security of the encrypted (Cipher) text resides in the key length and decryption process is a difficult without proper knowledge of the key. In sense while encryption a plain text using a secret algorithm, and sent to a second party who can decrypt the plain text back because they also has access to this secret algorithm.
where as, hashing is refer as one-way functions that compress arbitrary length strings into fixed short strings. Hash Functions can be designed using block ciphers using a secret key as a parameter along with the message that has to be hashed. The important thing and focus point about a hash value is that it is nearly impossible for any one to derive the original input value without knowing the data used to create the hash value. so we can see Hash working is not like encryption because Encrypting is a proper two-way function. It's reversible, you can decrypt the mangled string to get original string if you have the key which is not possible in Hash.

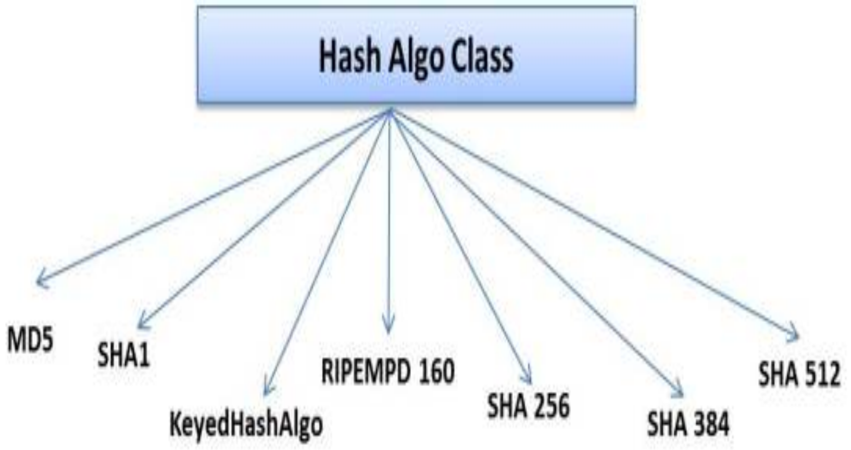


Animated Pic – How Encryption and Hash are different

For more Info, have a read
[Fundamental difference between Hashing and Encryption algorithms-StackOverFlow QuestionAnswer](#)
[Hashes are "digests", not "encryption"](#)

Hash Algorithm Classes

The .Net Framework provides some great classes to implement Hashing. The HashAlgorithm class providesbase functionality for all hashing classes in the .Net Framework. The HashAlgorithm class is an abstract(MustInherit) class that is extended by the SHA1, MD, KeyedHashAlgorithm, RIPEMD160, SHA 256,SHA 384and SHA 512 classes where as each of the class is sued to hash data



Hashing by using MD5 class

The MD5 class is an abstract(means must inherit) class which responsible to provides hashing functionality by using MD5 hashing algorithm.

What is MD5 class

The MD5 hashing algorithm is one of the two most commonly used hashing algorithms. There are known flaws in the algorithm and its known as its great features and its popularity.
in .Net Framework, the MD5 hashing algorithm class uses a 128-bit hash key and is extended by the MD5CryptoServiceProvder class which is also known as CSP class.

What is MD5CryptoServiceProvder(CSP) class

The MDCryptoServiceProvieder class extends the MD5 class. The MDCryptoServiceProvieder class that provides cryptographic service by wrapping unmanaged object that are external to the CLR(Common Language Runtime)

Implementation of MDCryptoServiceProvieder class

Here given code show you how to use MD5CryptoServiceProvider for hashing your data. GetMD5Hash is a function which need values as parameter. In function MD5provider is an instance of MD5CryptoServiceProvider. 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 MD5CryptoServiceProvider class you must need to mention System.Security.Cryptography Namespace*

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

Now create a function

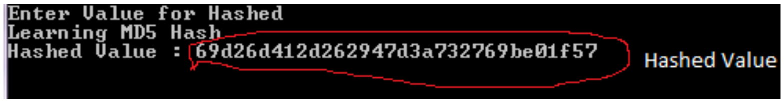
```
public static string GetMD5Hash(string plaintext)
        {
            MD5CryptoServiceProvider MD5provider = new MD5CryptoServiceProvider();
            byte[] hasedvalue = MD5provider.ComputeHash(Encoding.Default.GetBytes(plaintext));
            StringBuilder str = new StringBuilder();
            for (int counter = 0; counter < hasedvalue.Length; counter++)
            {
                str.Append(hasedvalue[counter].ToString("x2"));
            }
            return str.ToString();
        }
```

How to use Function

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

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

           string strhashed = GetMD5Hash(yourvalue);
           Console.WriteLine("Hashed Value : " + strhashed);
      }
```

What Output comes



```
Enter Value for Hashed
Learning MD5 Hash
Hashed Value : 69d26d412d262947d3a732769be01f57
```

How to Verify Hash

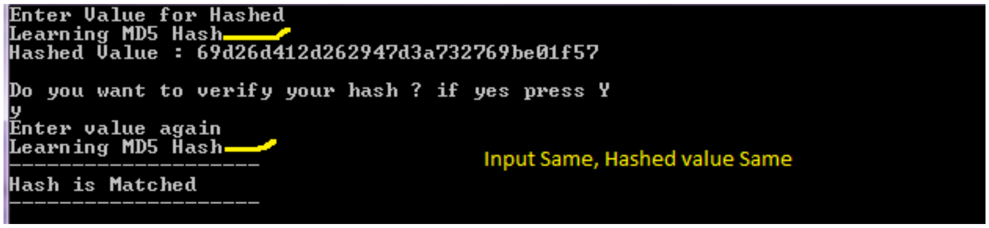
To Verify a hash against a string we can use StringComparer class which responsible to represents a string comparison operation that uses specific case and culture-based or ordinal comparison rules.

```
static bool VerifyMD5hash(string PlainText, string prevhashedvalue)
        {

            string hashedvalue2 = GetMD5Hash(PlainText);

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

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

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

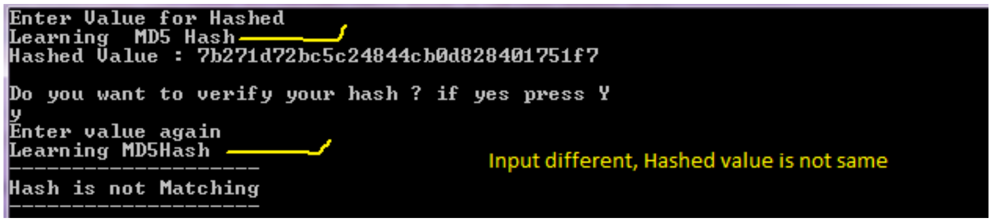
```
static void Main(string[] args)
        {
            Console.WriteLine("Enter Value for Hashed");
            string yourvalue = Console.ReadLine();

            string strhashed = GetMD5Hash(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 = VerifyMD5hash(yourvalue2, strhashed);
            Console.WriteLine("--------------------");
            if (res)
            {

                Console.WriteLine("Hash is Matched");

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

What output comes



```
Enter Value for Hashed
Learning MD5 Hash
Hashed Value : 69d26d412d262947d3a732769be01f57

Do you want to verify your hash ? if yes press Y
y
Enter value again
Learning MD5 Hash
--------------------
Hash is Matched
--------------------
```

Input Same, Hashed value Same



```
Enter Value for Hashed
Learning  MD5 Hash
Hashed Value : 7b271d72bc5c24844cb0d828401751f7

Do you want to verify your hash ? if yes press Y
y
Enter value again
Learning MD5Hash
--------------------
Hash is not Matching
--------------------
```

Input different, Hashed value is not same

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]
Don't Hash Secret

Coming Next

In the next article, I will hopefully come back with more basic details and implementation of SHA1 and other Hash Algo classes for beginners. Till then Happy Coding…

Share this:

Twitter Facebook Google Tumblr Reddit Email Print

Loading...

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. *Create a free website or blog at WordPress.com.*