**Step1:**

* 1. Open Visual Studio
  2. File >> New >> Project
  3. Select Windows from left panel
  4. Select class library and give appropriate name
  5. Change name Class1.cs to Encryptor
  6. Make it partial i.e. public partial class Encryptor
  7. Copy and paste following code into class

[SqlFunction()]

public static string Encrypt(string Input, string Password, string Salt)

{

if (Input == null || Input.Length <= 0) return "";

Rfc2898DeriveBytes keyGen = \_\_createKeyGen(Password, Salt);

ICryptoTransform transformer = \_\_createEncryptor(keyGen);

byte[] transformed = \_\_transform(Encoding.Default.GetBytes(Input), transformer);

return Convert.ToBase64String(transformed);

}

[SqlFunction()]

public static string Decrypt(string Input, string Password, string Salt)

{

if (Input == null || Input.Length <= 0) return "";

Rfc2898DeriveBytes keyGen = \_\_createKeyGen(Password, Salt);

ICryptoTransform transformer = \_\_createDecryptor(keyGen);

byte[] transformed = \_\_transform(Convert.FromBase64String(Input), transformer);

return Encoding.Default.GetString(transformed);

}

[SqlFunction()]

public static string Hash(string Input)

{

if (Input == null || Input.Length <= 0) return "";

StringBuilder result = new StringBuilder();

SHA1 provider = SHA1.Create();

byte[] \_\_result = provider.ComputeHash(Encoding.Default.GetBytes(Input));

foreach (Byte b in \_\_result)

result.Append(String.Format("{0:x2}", b));

return result.ToString();

}

private static Rfc2898DeriveBytes \_\_createKeyGen(string Password, string Salt)

{

return new Rfc2898DeriveBytes(Password, Encoding.Default.GetBytes(Salt));

}

private static ICryptoTransform \_\_createEncryptor(Rfc2898DeriveBytes KeyGen)

{

TripleDES provider = TripleDES.Create();

return provider.CreateEncryptor(KeyGen.GetBytes(16), KeyGen.GetBytes(16));

}

private static ICryptoTransform \_\_createDecryptor(Rfc2898DeriveBytes KeyGen)

{

TripleDES provider = TripleDES.Create();

return provider.CreateDecryptor(KeyGen.GetBytes(16), KeyGen.GetBytes(16));

}

private static byte[] \_\_transform(byte[] Input, ICryptoTransform Transformer)

{

MemoryStream ms = new MemoryStream();

byte[] result;

CryptoStream writer = new CryptoStream(ms, Transformer, CryptoStreamMode.Write);

writer.Write(Input, 0, Input.Length);

writer.FlushFinalBlock();

ms.Position = 0;

result = ms.ToArray();

ms.Close();

writer.Close();

return result;

}

**Step2:**

Now build the project

Copy SqlEncryptor.dll from following path:

‘SqlEncryptor\bin\Debug’

Paste it on different folder where sql server is installed.

**Step3:**

Open Sql Server

Create New Database



Give appropriate Name



Now go to Programmability >> Assemblies

Right click on assemblies and select New Assembly…



Select ‘SqlEncrptor.dll’ from location



Press ‘OK’ button and dll will be registered.

Run following code snippet to enable clr on sql

sp\_configure 'clr enabled', 1;

GO

reconfigure

GO

Create Encrypt and Decrypt function in DB

CREATE FUNCTION Encrypt

(@Input nvarchar(255), @Password nvarchar(255), @Salt nvarchar(255))

RETURNS nvarchar(255)

EXTERNAL NAME SqlEncryptor.Encryptor.Encrypt;

CREATE FUNCTION Decrypt

(@Input nvarchar(255), @Password nvarchar(255), @Salt nvarchar(255))

RETURNS nvarchar(255)

EXTERNAL NAME SqlEncryptor.Encryptor.Decrypt;

Now check encrypt and decrypt using c# function

DECLARE @PlainText VARCHAR(50)

DECLARE @EncryptedText VARCHAR(50)

SET @EncryptedText = (SELECT dbo.Encrypt('I am Akshay Patel', 'P@S$W0RD', '@K$H@YP@TEL'))

SET @PlainText= (SELECT dbo.Decrypt(@EncryptedText, 'P@S$W0RD', '@K$H@YP@TEL'))

SELECT @PlainText AS 'Plain Text'

SELECT @EncryptedText AS 'Encrypted Text'