Admin.aspx source:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="admin.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.style1

{

text-align: center;

font-family: Arial;

font-size: large;

}

.style2

{

width: 100%;

}

.style3

{

width: 473px;

text-align: right;

}

.style4

{

width: 473px;

text-align: right;

height: 30px;

}

.style5

{

height: 30px;

}

.style6

{

width: 469px;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<div class="style1" style="color: #FFFFFF; background-color: #0000FF"><strong>FUZZY SEARCHING OVER ENCRYPTED DATA IN CLOUD COMPUTING</strong></div><br />

<table class="style2">

<tr>

<td class="style3" style="text-align: right">

<strong>Enter File Name :</strong></td>

<td>

<asp:TextBox ID="TextBox1" runat="server" Height="24px" Width="154px"></asp:TextBox>

&nbsp;</td>

</tr>

<tr>

<td class="style3" style="text-align: right">

<strong>Enter keywords associated with the file

(SEPARATE BY SPACE):</strong> </td>

<td>

<asp:TextBox ID="TextBox2" runat="server" TextMode="MultiLine"></asp:TextBox>

&nbsp;<asp:RegularExpressionValidator ID="RegularExpressionValidator1"

runat="server" ErrorMessage="Separate using spaces"

ForeColor="Red"

ValidationExpression="^[a-zA-Z ]\*$"

ControlToValidate="TextBox2"></asp:RegularExpressionValidator>

</td>

</tr>

<tr>

<td class="style4">

<strong>Browse File :

</strong>

</td>

<td class="style5">

<asp:FileUpload ID="FileUpload1" runat="server" />

&nbsp;</td>

</tr>

<tr>

<td class="style3">

&nbsp;</td>

<td>

<asp:Button ID="Button2" runat="server" Text="Submit" onclick="Button2\_Click"

style="height: 26px" />

<asp:Label ID="Label1" runat="server" Text="Label" Visible="False"></asp:Label>

<br />

<br />

<asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>

</td>

</tr>

</table>

<br />

<table class="style2">

<tr>

<td class="style6" style="text-align: right">

<asp:TextBox ID="TextBox3" runat="server" Height="22px"

style="text-align: center; margin-left: 0px" Width="211px">Enter keywords to search</asp:TextBox>

</td>

<td style="text-align: left">

<asp:Button ID="Button3" runat="server" style="text-align: right"

Text="Search" />

</td>

</tr>

</table>

<br />

<br />

<br />

<br />

<br />

<br />

<br />

</div>

</form>

</body>

</html>

Admin.aspx.cs :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using MySql.Data.MySqlClient;

using System.Data;

using System.Net;

using System.IO;

using System.Collections.Specialized;

using System.Security.Cryptography;

using System.Text;

using Dropbox.Api.Auth;

using System.Data.SqlClient;

public partial class \_Default : System.Web.UI.Page

{

// private MySqlConnection connection;

// private string server;

// private string database;

// private string uid;

// private string password;

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\a\Documents\Visual Studio 2010\WebSites\Fuzzy\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

string passPhrase = "Pas5pr@se"; // can be any string

string saltValue = "s@1tValue"; // can be any string

string hashAlgorithm = "SHA1"; // can be "MD5"

int passwordIterations = 2; // can be any number

string initVector = "@1B2c3D4e5F6g7H8"; // must be 16 bytes

int keySize = 256; // can be 192 or 128

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["utype"] == null)

{

Response.Redirect("~/Login.aspx");

}

}

/\* protected void Button1\_Click(object sender, EventArgs e)

{

try

{

server = "192.168.1.35";

database = "test";

uid = "root";

password = "major";

string connectionString;

connectionString = "SERVER=" + server + ";" + "DATABASE=" +

database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

connection = new MySqlConnection(connectionString);

MySqlDataAdapter da = new MySqlDataAdapter("SELECT \* FROM sahil.test", connection);

connection.Open();

DataSet ds = new DataSet();

da.Fill(ds, "test");

GridView1.DataSource = ds;

GridView1.DataBind();

connection.Close();

}

catch (Exception ex)

{

Label1.Text = ex.ToString();

}

}

\*/

protected void Button2\_Click(object sender, EventArgs e)

{

// MySqlConnection con = new MySqlConnection("datasource=localhost;port=3306;username=root;password=");

// MySqlCommand cmd = new MySqlCommand("INSERT INTO fuzzy.test VALUES(@fn,@fl)", con);

// cmd.Parameters.AddWithValue("fn", TextBox1.Text);

// char a = Server.MapPath(FileUpload1.FileName);

// cmd.Parameters.AddWithValue("fl", a);

/\* TEST 1

\* WebClient webClient = new WebClient();

string sourceFilePath = @"C:\Users\a\Desktop\cet.txt";

string webAddress = "http://localhost/uploads";

string destinationFilePath = webAddress + "cet.txt";

//webClient.Credentials = new System.Net.NetworkCredential("username", "password");

webClient.UseDefaultCredentials = true;

byte[] responseArray = webClient.UploadFile(destinationFilePath, "POST", sourceFilePath);

webClient.Dispose(); \*/

/\* TEST 2

\* string fileToUpload = "C:\\Users\\a\\Desktop\\old.txt";

FileStream rdr = new FileStream(fileToUpload, FileMode.Open);

HttpWebRequest req = (HttpWebRequest)WebRequest.Create("http://localhost/uploads"); //Given URI exists

req.Method = "POST";

req.ContentLength = rdr.Length;

req.AllowWriteStreamBuffering = true;

Stream reqStream = req.GetRequestStream();

Console.WriteLine(rdr.Length);

byte[] inData = new byte[rdr.Length];

// Get data from upload file to inData

int bytesRead = rdr.Read(inData, 0, (int)rdr.Length);

// put data into request stream

reqStream.Write(inData, 0, (int)rdr.Length);

rdr.Close();

req.GetResponse();

// after uploading close stream

reqStream.Close();

\*/

/\*

NameValueCollection nvc = new NameValueCollection();

long length = 0;

string boundary = "----------------------------" +

DateTime.Now.Ticks.ToString("x");

HttpWebRequest httpWebRequest2 = (HttpWebRequest)WebRequest.Create("http://localhost/uploads");

httpWebRequest2.ContentType = "multipart/form-data; boundary=" +

boundary;

httpWebRequest2.Method = "POST";

httpWebRequest2.KeepAlive = true;

httpWebRequest2.Credentials =

System.Net.CredentialCache.DefaultCredentials;

Stream memStream = new System.IO.MemoryStream();

byte[] boundarybytes = System.Text.Encoding.ASCII.GetBytes("\r\n--" +

boundary + "\r\n");

string formdataTemplate = "\r\n--" + boundary +

"\r\nContent-Disposition: form-data; name=\"{0}\";\r\n{1}";

foreach (string key in nvc.Keys)

{

string formitem = string.Format(formdataTemplate, key, nvc[key]);

byte[] formitembytes = System.Text.Encoding.UTF8.GetBytes(formitem);

memStream.Write(formitembytes, 0, formitembytes.Length);

}

memStream.Write(boundarybytes, 0, boundarybytes.Length);

string headerTemplate = "Content-Disposition: form-data; name=\"{0}\"; filename=\"{1}\"\r\nContent-Type: text/plain\r\n";

string files = "C://Users//a//Desktop//cet.txt";

//string header = string.Format(headerTemplate, "file" + i, files[i]);

string header = string.Format(headerTemplate, "uplTheFile", files);

byte[] headerbytes = System.Text.Encoding.UTF8.GetBytes(header);

memStream.Write(headerbytes, 0, headerbytes.Length);

FileStream fileStream = new FileStream(files, FileMode.Open,

FileAccess.Read);

byte[] buffer = new byte[1024];

int bytesRead = 0;

while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) != 0)

{

memStream.Write(buffer, 0, bytesRead);

}

memStream.Write(boundarybytes, 0, boundarybytes.Length);

fileStream.Close();

httpWebRequest2.ContentLength = memStream.Length;

Stream requestStream = httpWebRequest2.GetRequestStream();

memStream.Position = 0;

byte[] tempBuffer = new byte[memStream.Length];

memStream.Read(tempBuffer, 0, tempBuffer.Length);

memStream.Close();

requestStream.Write(tempBuffer, 0, tempBuffer.Length);

requestStream.Close();

WebResponse webResponse2 = httpWebRequest2.GetResponse();

Stream stream2 = webResponse2.GetResponseStream();

StreamReader reader2 = new StreamReader(stream2);

// MessageBox.Show(reader2.ReadToEnd());

Label2.Text = reader2.ReadToEnd();

webResponse2.Close();

httpWebRequest2 = null;

webResponse2 = null;

\*/

//Upload(Server.MapPath(FileUpload1.FileName));

// FileUpload1.SaveAs(Server.MapPath("Uploadedfiles\\"+FileUpload1.FileName));

/\* FileUpload1.SaveAs(Server.MapPath(FileUpload1.FileName));

string a = Server.MapPath(FileUpload1.FileName);

string b = Server.MapPath(FileUpload1.FileName);

encrypt64(a, b);

Upload(b ); \*/

// SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\a\Documents\Visual Studio 2010\WebSites\Fuzzy\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand("INSERT INTO keywords VALUES(@fn,@key)",con);

cmd.Parameters.AddWithValue("fn", TextBox1.Text);

cmd.Parameters.AddWithValue("key", TextBox2.Text);

cmd.ExecuteNonQuery();

con.Close();

gen(TextBox2.Text,TextBox1.Text);

}

public void gen(string keyw,string fname)

{

string[] tokens = keyw.Split(' ');

for(int i = 0; i < tokens.Length; i++)

{

string[] grams = new string[50];

int v = 0;

for(int j = 0; j < tokens[i].Length - 2; j++)

{

grams[j] = tokens[i].Substring(v, 3);

v++;

}

foreach (string element in grams)

{

// string s = grams[k];

if (element == null)

break;

else

{

byte[] initVectorBytes = Encoding.ASCII.GetBytes(initVector);

byte[] saltValueBytes = Encoding.ASCII.GetBytes(saltValue);

byte[] plainTextBytes = Encoding.UTF8.GetBytes(element);

PasswordDeriveBytes password = new PasswordDeriveBytes(passPhrase, saltValueBytes, hashAlgorithm, passwordIterations);

byte[] keyBytes = password.GetBytes(keySize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.CBC;

ICryptoTransform encryptor = symmetricKey.CreateEncryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream();

CryptoStream cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write);

cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);

cryptoStream.FlushFinalBlock();

byte[] cipherTextBytes = memoryStream.ToArray();

memoryStream.Close();

cryptoStream.Close();

string cipherText = Convert.ToBase64String(cipherTextBytes);

con.Open();

SqlCommand cmd = new SqlCommand("INSERT INTO ngrams VALUES(@fn,@gr)", con);

cmd.Parameters.AddWithValue("fn", fname);

cmd.Parameters.AddWithValue("gr", cipherText);

cmd.ExecuteNonQuery();

con.Close();

/\* WORKING DECRYPTION

byte[] cipherTextBytes2 = Convert.FromBase64String("0dGGulalIu8VWx5HtW2u5A==");

ICryptoTransform decryptor = symmetricKey.CreateDecryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream2 = new MemoryStream(cipherTextBytes2);

CryptoStream cryptoStream2 = new CryptoStream(memoryStream2, decryptor, CryptoStreamMode.Read);

byte[] plainTextBytes2 = new byte[cipherTextBytes2.Length];

int decryptedByteCount = cryptoStream2.Read(plainTextBytes2,0,plainTextBytes2.Length);

memoryStream2.Close();

cryptoStream2.Close();

string plain = Encoding.UTF8.GetString(plainTextBytes2, 0, decryptedByteCount);

Label2.Text = plain; \*/

}

}

}

/\*for (int i = 0; i < (tokens.Length - 1); i++)

{

string s = "";

int start = i;

int end = i + 2;

for (int j = start; j < end; j++)

{

s = s + "" + tokens[j];

} \*/

/\* SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\a\Documents\Visual Studio 2010\WebSites\Fuzzy\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand("INSERT INTO ngrams VALUES(@fn,@ng)",con);

cmd.Parameters.AddWithValue("fn",fname);

cmd.Parameters.AddWithValue("ng",grams);

cmd.ExecuteNonQuery();

con.Close(); \*/

}

public void Upload(string Filetoup)

{

try

{

/\* // Get the object used to communicate with the server.

FtpWebRequest request = (FtpWebRequest)WebRequest.Create("ftp://127.0.0.1/");

request.Method = WebRequestMethods.Ftp.UploadFile;

// This example assumes the FTP site uses anonymous logon.

request.Credentials = new NetworkCredential("vish", "vish");

// Copy the entire contents of the file to the request stream.

StreamReader sourceStream = new StreamReader(Filetoup);

byte[] fileContents = Encoding.UTF8.GetBytes(sourceStream.ReadToEnd());

sourceStream.Close();

request.ContentLength = fileContents.Length;

// Upload the file stream to the server.

Stream requestStream = request.GetRequestStream();

requestStream.Write(fileContents, 0, fileContents.Length);

requestStream.Close();

// Get the response from the FTP server.

FtpWebResponse response = (FtpWebResponse)request.GetResponse();

// Close the connection = Happy a FTP server.

response.Close();

// Return the status of the upload.

// return response.StatusDescription; \*/

/\* APRIL'2016

FtpWebRequest myFtpWebRequest; //An object for a communication request with the FTP server i.e. ftp://127.0.0.1.

FtpWebResponse myFtpWebResponse; //.. For getting response from server

StreamWriter myStreamWriter; //For writing characters

string random = Path.GetRandomFileName(); //RANDOMIZE FILE NAME FOR SECURITY PURPOSES.

myFtpWebRequest = (FtpWebRequest)WebRequest.Create("ftp://127.0.0.1/" + random); //myFtpWebRequest = (FtpWebRequest)WebRequest.Create("ftp://127.0.0.1/"+FileUpload1.FileName);

myFtpWebRequest.Credentials = new NetworkCredential("vish", "vish"); //ftp uname:vish, pass:vish

myFtpWebRequest.Method = WebRequestMethods.Ftp.UploadFile; //For uploading a file

myFtpWebRequest.UseBinary = true; //Data to be transferred is in binary. FOR IMAGE - TRUE, FOR TEXT - FALSE

myStreamWriter = new StreamWriter(myFtpWebRequest.GetRequestStream()); //Get the stream object to write data.

myStreamWriter.Write(new StreamReader(Server.MapPath(FileUpload1.FileName)).ReadToEnd()); // Read characters from byte stream

myStreamWriter.Close();

myFtpWebResponse = (FtpWebResponse)myFtpWebRequest.GetResponse();

Response.Write("Upload File Complete, status: " + myFtpWebResponse.StatusDescription);

myFtpWebResponse.Close(); \*/

}

catch(WebException e)

{

// Label1.Text = ((FtpWebResponse)e.Response).StatusDescription;

}

}

public void encrypt64(string inputFile, string outputFile) /\* Convert file to be uploaded into bytes, then convert those bytes

into base64 string, then overwrite the file with the string\*/

{

byte[] AsBytes = File.ReadAllBytes(inputFile);

string base64 = Convert.ToBase64String(AsBytes);

File.WriteAllText(outputFile, base64);

}

private void GetAccessToken()

{

}

public void dropboxup()

{

}

}

Login.aspx:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.style1

{

text-align: center;

font-size: large;

font-family: Arial;

}

#form1

{

height: 28px;

}

.style2

{

width: 100%;

}

.style3

{

width: 465px;

text-align: right;

}

.style4

{

width: 465px;

height: 23px;

}

.style5

{

height: 23px;

}

.style6

{

width: 465px;

height: 23px;

text-align: right;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div class="style1" style="background-color: #0000FF; color: #FFFFFF;"><strong>FUZZY SEARCHING OVER ENCRYPTED DATA IN CLOUD COMPUTING</strong></div><br />

<table class="style2">

<tr>

<td class="style3">

Username :</td>

<td>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td class="style6">

Password :</td>

<td class="style5">

<asp:TextBox ID="TextBox2" runat="server" TextMode="Password"></asp:TextBox>

</td>

</tr>

<tr>

<td class="style6">

User type :</td>

<td class="style5">

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem Value="a">Admin</asp:ListItem>

<asp:ListItem Value="o">Other</asp:ListItem>

</asp:DropDownList>

</td>

</tr>

<tr>

<td class="style4">

&nbsp;</td>

<td class="style5">

<asp:Button ID="Button1" runat="server" Text="Submit" onclick="Button1\_Click" />

<br />

<asp:Label ID="Label1" runat="server" Font-Bold="True" ForeColor="Red"

Text="WRONG CREDENTIALS!" Visible="False"></asp:Label>

</td>

</tr>

</table>

</form>

<p>

&nbsp;</p>

<p>

&nbsp;</p>

<p>

&nbsp;</p>

<p>

&nbsp;</p>

</body>

</html>

Login.aspx.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

public partial class Login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Label1.Visible = false;

}

protected void Button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\a\Documents\Visual Studio 2010\WebSites\Fuzzy\App\_Data\Database.mdf;Integrated Security=True;User Instance=True");

con.Open();

SqlCommand cmd = new SqlCommand("SELECT \* FROM Master\_Login WHERE uname=@u AND password=@p AND utype=@ut", con);

cmd.Parameters.AddWithValue("u", TextBox1.Text);

cmd.Parameters.AddWithValue("p", TextBox2.Text);

cmd.Parameters.AddWithValue("ut", DropDownList1.SelectedValue.ToString());

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

cmd.ExecuteReader();

con.Close();

if (dt != null && dt.Rows.Count > 0) //If login table not empty then do the following

{

string utype = DropDownList1.SelectedValue.ToString();

Session["utype"] = DropDownList1.SelectedItem.ToString();

if (utype == "a")

{

Response.Redirect("~/admin.aspx");

}

else if (utype == "o")

{

Response.Redirect("~/user.aspx");

}

}

else

{

Label1.Visible = true;

}

}

}

User.aspx:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="user.aspx.cs" Inherits="user" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.style1

{

text-align: center;

font-family: Arial;

font-size: large;

}

.style2

{

width: 100%;

}

.style3

{

width: 438px;

text-align: right;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<div class="style1" style="color: #FFFFFF; background-color: #0000FF"><strong>FUZZY SEARCHING OVER ENCRYPTED DATA IN CLOUD COMPUTING</strong></div><br />

</div>

<table class="style2">

<tr>

<td class="style3">

<asp:TextBox ID="TextBox1" runat="server" Height="22px"

style="text-align: center; margin-left: 0px" Width="211px">Enter keywords to search</asp:TextBox>

</td>

<td>

<asp:Button ID="Button1" runat="server" style="text-align: right"

Text="Search" onclick="Button1\_Click" />

<asp:RegularExpressionValidator ID="RegularExpressionValidator1"

runat="server" ErrorMessage="Separate using spaces"

ForeColor="Red"

ValidationExpression="^[a-zA-Z ]\*$"

ControlToValidate="TextBox1"></asp:RegularExpressionValidator>

</td>

</tr>

</table>

</form>

</body>

</html>

User.aspx.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Security.Cryptography;

using System.Text;

using System.IO;

public partial class user : System.Web.UI.Page

{

string passPhrase = "Pas5pr@se"; // can be any string

string saltValue = "s@1tValue"; // can be any string

string hashAlgorithm = "SHA1"; // can be "MD5"

int passwordIterations = 2; // can be any number

string initVector = "@1B2c3D4e5F6g7H8"; // must be 16 bytes

int keySize = 256; // can be 192 or 128

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["utype"] == null)

{

Response.Redirect("~/Login.aspx");

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

gen(TextBox1.Text);

}

public void gen(string keyw)

{

string[] tokens = keyw.Split(' ');

for (int i = 0; i < tokens.Length; i++)

{

string[] grams = new string[50];

int v = 0;

for (int j = 0; j < tokens[i].Length - 2; j++)

{

grams[j] = tokens[i].Substring(v, 3);

v++;

}

foreach (string element in grams)

{

// string s = grams[k];

if (element == null)

break;

else

{

byte[] initVectorBytes = Encoding.ASCII.GetBytes(initVector);

byte[] saltValueBytes = Encoding.ASCII.GetBytes(saltValue);

byte[] plainTextBytes = Encoding.UTF8.GetBytes(element);

PasswordDeriveBytes password = new PasswordDeriveBytes(passPhrase, saltValueBytes, hashAlgorithm, passwordIterations);

byte[] keyBytes = password.GetBytes(keySize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.CBC;

ICryptoTransform encryptor = symmetricKey.CreateEncryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream();

CryptoStream cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write);

cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);

cryptoStream.FlushFinalBlock();

byte[] cipherTextBytes = memoryStream.ToArray();

memoryStream.Close();

cryptoStream.Close();

string cipherText = Convert.ToBase64String(cipherTextBytes);

}

}

}

}

}

Three tables: kewords [FileName varchar(50), Keywords(100)]

, Master\_Login [uname varchar(50), password varchar(50), utype char(1)]

, ngrams [Fname varchar(100), gram varchar(MAX).]