

Forum:[Hack Forums](#)

Thread: [\[C#\] XTEA Encryption Source](#)

Mr. Anderson - November 13, 2010, 10:36 am

All credits to decompiler for making the original encryption, I just ported it to C#:
(There might be 1 or two errors I havnt found yet in the C# version. If so please post it.

Code using Microsoft.VisualBasic;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Diagnostics;

public class XTea

{

 //Leave a thanks....

 //- decompiler

 //ported by mr. anderson

 private static uint[] f_key;

 public XTea(string password)

 {

 f_key = FormatKey(password);

 }

 public static string Encrypt(string input)

 {

 if (input.Length == 0) {

 throw new ArgumentException("Invalid input length!");

 }

 if (input.Length % 2 != 0) {

 input += ControlChars.NullChar;

 }

 byte[] dataBytes = System.Text.ASCIIEncoding.ASCII.GetBytes(input);

 string cipher = string.Empty;

 uint[] tempData = new uint[2];

 for (int i = 0; i <= dataBytes.Length - 1; i += 2) {

 tempData[0] = dataBytes[i];

 tempData[1] = dataBytes[i + 1];

 code(tempData, f_key);

 cipher += ConvertUIntToString(tempData[0]) + ConvertUIntToString(tempData[1]);

 }

 return cipher;

 }

```

public static string Decrypt(string Data)
{
    int x = 0;
    uint[] tempData = new uint[2];
    byte[] dataBytes = new byte[(Data.Length / 8) * 2];
    for (int i = 0; i <= Data.Length - 1; i += 8) {
        tempData[0] = ConvertStringToUInt(Data.Substring(i, 4));
        tempData[1] = ConvertStringToUInt(Data.Substring(i + 4, 4));
        decode(tempData, f_key);
        dataBytes[x] = Convert.ToByte(tempData[0]);
        x += 1;
        dataBytes[x] = Convert.ToByte(tempData[1]);
        x += 1;
    }
    string decipheredString = System.Text.ASCIIEncoding.ASCII.GetString(dataBytes, 0,
dataBytes.Length);
    if (decipheredString[decipheredString.Length - 1] == ControlChars.NullChar) {
        decipheredString = decipheredString.Substring(0, decipheredString.Length - 1);
    }
    return decipheredString;
}

private static uint[] FormatKey(string Key)
{
    if (Key.Length <= 5) {
        throw new ArgumentException("Key must be between 6 and 16 characters in length");
    }

    Key = Key.PadRight(16, ' ').Substring(0, 16);

    uint[] formattedKey = new uint[5];

    int j = 0;
    for (int i = 0; i <= Key.Length - 1; i += 4) {
        formattedKey[j] = ConvertStringToUInt(Key.Substring(i, 4));
        j += 1;
    }
    return formattedKey;
}

private static void code(uint[] v, uint[] k)
{
    uint y = v[0];
    uint z = v[1];
    uint sum = 0;
    uint delta = 0x9e3779b9u;

    for (i = 0; i <= 31; i++) {
        y += (z << 4 ^ z >> 5) + z ^ sum + k[sum & 3];
        sum += delta;
        z += (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3];
    }
}

```

```

    v[0] = y;
    v[1] = z;
}

private static void decode(uint[] v, uint[] k)
{
    uint y = v[0];
    uint z = v[1];
    uint sum = 0xc6ef3720u;
    uint delta = 0x9e3779b9u;

    for (i = 0; i <= 31; i++) {
        z -= (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3];
        sum -= delta;
        y -= (z << 4 ^ z >> 5) + z ^ sum + k[sum & 3];
    }

    v[0] = y;
    v[1] = z;
}

private static System.UInt32 ConvertStringToUInt(string Input)
{
    UInt32 output = default(UInt32);
    output = Convert.ToUInt32(FileSystem.Input(0));
    output += (Convert.ToUInt32(FileSystem.Input(1)) << 8);
    output += (Convert.ToUInt32(FileSystem.Input(2)) << 16);
    output += (Convert.ToUInt32(FileSystem.Input(3)) << 24);
    return output;
}

private static string ConvertUIntToString(System.UInt32 Input)
{
    System.Text.StringBuilder output = new System.Text.StringBuilder();
    output.Append(Convert.ToChar(Input & 0xff));
    output.Append(Convert.ToChar((Input >> 8) & 0xff));
    output.Append(Convert.ToChar((Input >> 16) & 0xff));
    output.Append(Convert.ToChar((Input >> 24) & 0xff));
    return output.ToString();
}
}

```

Example:

[img]http://ploader.net/files/743101b5744d2c56f961988109d931e8.gif[/img]

Grieve™ - November 13, 2010, 11:58 am

looks nice! i will try later. good job converting it to c#

TheHacker221 - November 13, 2010, 4:31 pm

Bad conversion to C#. I would recommend this to ANYONE winning to convert byte arrays, strings, ints and stuff to use this. I will be using this aswell I have fixed the code up for C# as thats my primary programming language

Code using Microsoft.VisualBasic;

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Diagnostics;
using System.Text;

public class XTea
{
    private static uint[] f_key;
    public XTea(string password)
    {
        f_key = FormatKey(password);
    }

    public static string Encrypt(string input)
    {
        if (input.Length == 0)
        {
            throw new ArgumentException("Invalid input length!");
        }

        if (input.Length % 2 != 0)
        {
            input += "\0";
        }

        byte[] dataBytes = ASCIIEncoding.ASCII.GetBytes(input);

        string cipher = string.Empty;
        uint[] tempData = new uint[2];
        for (int i = 0; i <= dataBytes.Length - 1; i += 2)
        {
            tempData[0] = dataBytes[i];
            tempData[1] = dataBytes[i + 1];
            code(tempData, f_key);
            cipher += ConvertUIntToString(tempData[0]) + ConvertUIntToString(tempData[1]);
        }

        return cipher;
    }

    public static string Decrypt(string Data)
    {
        int x = 0;
        uint[] tempData = new uint[2];
        byte[] dataBytes = new byte[(Data.Length / 8) * 2];
        for (int i = 0; i <= Data.Length - 1; i += 8)
        {
            tempData[0] = ConvertStringToUInt(Data.Substring(i, 4));
            tempData[1] = ConvertStringToUInt(Data.Substring(i + 4, 4));
            decode(tempData, f_key);
            dataBytes[x] = Convert.ToByte(tempData[0]);
        }
    }
}

```

```

    x += 1;
    dataBytes[x] = Convert.ToByte(tempData[1]);
    x += 1;
}
string decipheredString = System.Text.ASCIIEncoding.ASCII.GetString(dataBytes, 0,
dataBytes.Length);
if (decipheredString[decipheredString.Length - 1] == Convert.ToChar("\0"))
{
    decipheredString = decipheredString.Substring(0, decipheredString.Length - 1);
}
return decipheredString;
}

```

```

private static uint[] FormatKey(string Key)
{
    if (Key.Length <= 5)
    {
        throw new ArgumentException("Key must be between 6 and 16 characters in length");
    }
}

```

```

Key = Key.PadRight(16, ' ').Substring(0, 16);

```

```

uint[] formattedKey = new uint[5];

```

```

int j = 0;
for (int i = 0; i <= Key.Length - 1; i += 4)
{
    formattedKey[j] = ConvertStringToUInt(Key.Substring(i, 4));
    j += 1;
}
return formattedKey;
}

```

```

private static void code(uint[] v, uint[] k)
{
    uint y = v[0];
    uint z = v[1];
    uint sum = 0;
    uint delta = 0x9e3779b9u;

    for (int i = 0; i <= 31; i++)
    {
        y += (z << 4 ^ z >> 5) + z ^ sum + k[sum & 3];
        sum += delta;
        z += (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3];
    }
}

```

```

v[0] = y;
v[1] = z;
}

```

```

private static void decode(uint[] v, uint[] k)
{

```

```

uint y = v[0];
uint z = v[1];
uint sum = 0xc6ef3720u;
uint delta = 0x9e3779b9u;

for (int i = 0; i <= 31; i++)
{
    z -= (y << 4 ^ y >> 5) + y ^ sum + k[sum >> 11 & 3];
    sum -= delta;
    y -= (z << 4 ^ z >> 5) + z ^ sum + k[sum & 3];
}

v[0] = y;
v[1] = z;
}

private static UInt32 ConvertStringToUInt(string Input)
{
    UInt32 output = default(UInt32);
    output = Input[0];
    output += (uint)(Input[1] << 8);
    output += (uint)(Input[2] << 16);
    output += (uint)(Input[3] << 24);

    return output;
}

private static string ConvertUIntToString(UInt32 Input)
{
    StringBuilder output = new StringBuilder();
    output.Append((char)Input & 0xff);
    output.Append((char)(Input >> 8) & 0xff);
    output.Append((char)(Input >> 16) & 0xff);
    output.Append((char)(Input >> 24) & 0xff);
    return output.ToString();
}
}

```

DarkSeven46 - November 13, 2010, 5:31 pm

Very nice, Will add to my brain.