Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Криптография и информационная безопасность”

Лабораторная работа 2

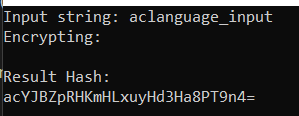
Проверил: Чербу Ольга

Выполнил: Чобану Артём

Кишинев 2022

**Алгоритм SHA1**

Шифруемое слово: **aclanguage\_inpu**

Скриншоты работы программы:

Код программы:

using System.Numerics;  
using System.Text;  
  
var input = "aclanguage\_input";  
Console.WriteLine($"Input string: {input}");  
  
var inputBytes = Encoding.UTF8.GetBytes(input);  
  
Console.WriteLine("Encrypting:");  
var encrypted = Encrypt(inputBytes.ToList());  
  
Console.WriteLine("\nResult Hash:");  
Console.WriteLine(Convert.ToBase64String(encrypted));  
  
Console.ReadKey();  
  
byte[] Encrypt(List<byte> inputBytes)  
{  
 uint A = 0x67452301;  
 uint B = 0xEFCDAB89;  
 uint C = 0x98BADCFE;  
 uint D = 0x10325476;  
 uint E = 0xC3D2E1F0;  
  
 inputBytes.Add(1);  
 while ((inputBytes.Count \* 8) % 512 < 448)  
 {  
 inputBytes.Add(0);  
 }  
 inputBytes.AddRange(BitConverter.GetBytes((ulong)inputBytes.Count));

for (int x = 0; x < inputBytes.Count; x += 512 / 8)  
 {  
 var w = ByteArrayToIntArray(inputBytes.GetRange(x, 512 / 8).ToArray()).ToList();  
  
 for (int i = 16; i <= 79; i++)  
 {  
 var wi = w[i - 3] ^ w[i - 8] ^ w[i - 14] ^ w[i - 16];  
  
 w.Add(BitOperations.RotateLeft(wi, 1));  
 }  
  
 uint a = A;  
 uint b = B;  
 uint c = C;  
 uint d = D;  
 uint e = E;  
  
 for (int i = 0; i <= 79; i++)  
 {  
 uint f, k;  
  
 if (i is >= 0 and <= 19)  
 {  
 f = (b & c) | (~b & d);  
 k = 0x5A827999;  
 }  
 else if (i is >= 20 and <= 39)  
 {  
 f = b ^ c ^ d;  
 k = 0x6ED9EBA1;  
 }  
 else if (i is >= 40 and <= 59)  
 {  
 f = (b & c) | (b & d) | (c & d);  
 k = 0x8F1BBCDC;  
 }  
 else   
 {  
 f = b ^ c ^ d;  
 k = 0xCA62C1D6;  
 }  
  
 var temp = BitOperations.RotateLeft(a, 5) + f + e + k + w[i];  
 e = d;  
 d = c;  
 c = BitOperations.RotateLeft(b, 30);  
 b = a;  
 a = temp;  
 }  
  
 A += a;  
 B += b;  
 C += c;  
 D += d;  
 E += e;  
 }  
  
 var result = new[] { A, B, C, D, E };  
 return IntArrayToByteArray(result);  
}  
  
static byte[] IntArrayToByteArray(uint[] integerArray)  
{  
 byte[] result = new byte[integerArray.Length \* 4];  
  
 for (int i = 0; i < result.Length; i++)  
 {  
 result[i] = (byte)((integerArray[i / 4] >> i % 4 \* 8) & 0xff);  
 }  
  
 return result;  
}  
  
static uint[] ByteArrayToIntArray(byte[] bytes)  
{  
 uint[] result = new uint[bytes.Length / 4];  
  
 for (int i = 0, counter = 0; i < result.Length; i++)  
 {  
 var current = (bytes[counter++] & 0xff) |  
 ((bytes[counter++] & 0xff) << 8) |  
 ((bytes[counter++] & 0xff) << 16) |  
 ((bytes[counter++] & 0xff) << 24);  
  
 result[i] = (uint)current;  
 }  
  
 return result;  
}

Выводы:

Алгоритм SHA1 является самым распространённым среди всех алгоритмов семейства SHA. Он используется в: Git, Mercurial, SSL, SSH и в других местах.

Многие компании отказываются от данного алгоритма с 2014 года.