Основи інформаційної безпеки

ПЗ №3

Тема: “Хеш функції та перевірка цілісності інформації”

Підготував:

Студент групи МІТ-21

Пономаренко Владислав

(КОД)

(З-1)

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3

{

class Program

{

static byte[] ComputeHashMd5(byte[] dataForHash)

{

using (var md5 = MD5.Create())

{

return md5.ComputeHash(dataForHash);

}

}

public static byte[] ComputeHashSha1(byte[] toBeHashed)

{

using (var sha1 = SHA1.Create())

{

return sha1.ComputeHash(toBeHashed);

}

}

public static byte[] ComputeHashSha256(byte[] toBeHashed)

{

using (var sha256 = SHA256.Create())

{

return sha256.ComputeHash(toBeHashed);

}

}

public static byte[] ComputeHashSha384(byte[] toBeHashed)

{

using (var sha384 = SHA384.Create())

{

return sha384.ComputeHash(toBeHashed);

}

}

public static byte[] ComputeHashSha512(byte[] toBeHashed)

{

using (var sha512 = SHA512.Create())

{

return sha512.ComputeHash(toBeHashed);

}

}

static void Main(string[] args)

{

const string strForHash1 = "Hello World!";

const string strForHash2 = "Hello world!";

const string strForHash3 = "Hello world!";

var md5ForStr1 = ComputeHashMd5(Encoding.Unicode.GetBytes(strForHash1));

var md5ForStr2 = ComputeHashMd5(Encoding.Unicode.GetBytes(strForHash2));

var md5ForStr3 = ComputeHashMd5(Encoding.Unicode.GetBytes(strForHash3));

Guid guidForMD\_1 = new Guid(md5ForStr1),

guidForMD\_2 = new Guid(md5ForStr2),

guidForMD\_3 = new Guid(md5ForStr3);

Console.WriteLine("MD5:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

Console.WriteLine(Convert.ToBase64String(md5ForStr1));

Console.WriteLine(Convert.ToBase64String(md5ForStr2));

Console.WriteLine(Convert.ToBase64String(md5ForStr3));

var sha1ForStr1 = ComputeHashSha1(Encoding.Unicode.GetBytes(strForHash1));

var sha1ForStr2 = ComputeHashSha1(Encoding.Unicode.GetBytes(strForHash2));

var sha1ForStr3 = ComputeHashSha1(Encoding.Unicode.GetBytes(strForHash3));

Console.WriteLine("SHA1:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

Console.WriteLine(Convert.ToBase64String(sha1ForStr1));

Console.WriteLine(Convert.ToBase64String(sha1ForStr2));

Console.WriteLine(Convert.ToBase64String(sha1ForStr3));

var sha256ForStr1 = ComputeHashSha256(Encoding.Unicode.GetBytes(strForHash1));

var sha256ForStr2 = ComputeHashSha256(Encoding.Unicode.GetBytes(strForHash2));

var sha256ForStr3 = ComputeHashSha256(Encoding.Unicode.GetBytes(strForHash3));

Console.WriteLine("SHA256:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

Console.WriteLine(Convert.ToBase64String(sha256ForStr1));

Console.WriteLine(Convert.ToBase64String(sha256ForStr2));

Console.WriteLine(Convert.ToBase64String(sha256ForStr3));

var sha384ForStr1 = ComputeHashSha384(Encoding.Unicode.GetBytes(strForHash1));

var sha384ForStr2 = ComputeHashSha384(Encoding.Unicode.GetBytes(strForHash2));

var sha384ForStr3 = ComputeHashSha384(Encoding.Unicode.GetBytes(strForHash3));

Console.WriteLine("SHA384:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

Console.WriteLine(Convert.ToBase64String(sha384ForStr1));

Console.WriteLine(Convert.ToBase64String(sha384ForStr2));

Console.WriteLine(Convert.ToBase64String(sha384ForStr3));

var sha512ForStr1 = ComputeHashSha512(Encoding.Unicode.GetBytes(strForHash1));

var sha512ForStr2 = ComputeHashSha512(Encoding.Unicode.GetBytes(strForHash2));

var sha512ForStr3 = ComputeHashSha512(Encoding.Unicode.GetBytes(strForHash3));

Console.WriteLine("SHA512:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

Console.WriteLine(Convert.ToBase64String(sha512ForStr1));

Console.WriteLine(Convert.ToBase64String(sha512ForStr2));

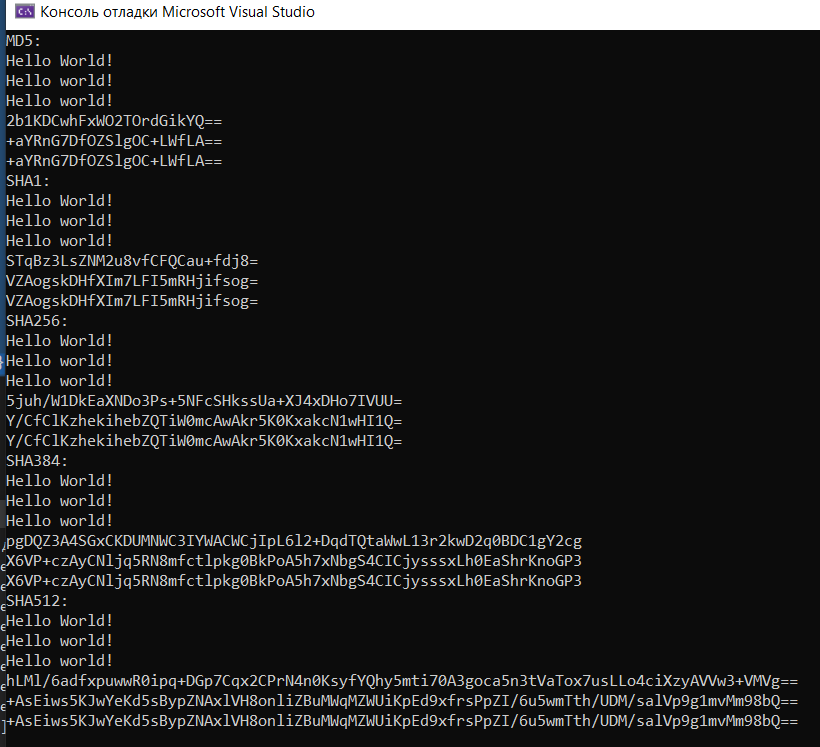
Console.WriteLine(Convert.ToBase64String(sha512ForStr3));

}

}

}

(РЕЗУЛЬТАТ)



(З-2)

(КОД)

using System;

using System.Text;

using System.Security.Cryptography;

namespace lab3x2

{

class Program

{

static byte[] Md5(byte[] DataForHash)

{

using (var md5 = MD5.Create())

{

return md5.ComputeHash(DataForHash);

}

}

static void Main(string[] args)

{

Guid guid = new Guid("564c8da6-0440-88ec-d453-0bbad57c6036");

Console.WriteLine(guid);

int mem;

for (int i = 100000000; i < 200000003; i++)

{

var md5 = Md5(Encoding.Unicode.GetBytes(i.ToString().Substring(1, 8)));

if (new Guid(md5) == guid)

{

Console.WriteLine("Пароль= " + i);

Console.WriteLine(Convert.ToBase64String(md5));

mem = i;

}

}

}

}

}

(РЕЗУЛЬТАТ)



(З-3)

(КОД)

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3x3

{

class Program

{

static byte[] ComputeHashSHA256(byte[] DataForHash)

{

using (var sha256 = SHA256.Create())

{

return sha256.ComputeHash(DataForHash);

}

}

public static byte[] ComputeHmacsha256(byte[] ToBeHashed, byte[] key)

{

using (var hmac = new HMACSHA256(key))

{

return hmac.ComputeHash(ToBeHashed);

}

}

static void Main(string[] args)

{

string key = "Vladpon";

string message = "love is a feeling";

var Key = ComputeHashSHA256(Encoding.Unicode.GetBytes(key));

var Mess = ComputeHmacsha256(Encoding.Unicode.GetBytes(message), Key);

Console.WriteLine($"Сообщение: {message}");

Console.WriteLine($"Хешированное сообщение: {Convert.ToBase64String(Mess)}");

Console.WriteLine("Отправка сообщения получателю....");

string key2 = "Vladpon";

var Key2 = ComputeHashSHA256(Encoding.Unicode.GetBytes(key2));

var Mess2 = ComputeHmacsha256(Encoding.Unicode.GetBytes(message), Key2);

if (Convert.ToBase64String(Mess) == Convert.ToBase64String(Mess2))

{

Console.WriteLine("Сообщение верное!");

}

else

{

Console.WriteLine("Сообщение не верное!");

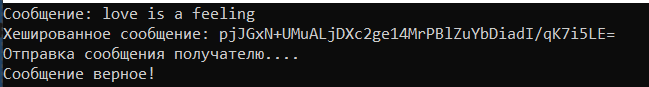
}

}

}

}

(РЕЗУЛЬТАТ)



(З-4)

(КОД)

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3x4

{

class Program

{

static byte[] ComputeHashSHA256(byte[] DataForHash)

{

using (var sha256 = SHA256.Create())

{

return sha256.ComputeHash(DataForHash);

}

}

static void Main(string[] args)

{

Console.WriteLine("Для регистрации введите логин и пароль.");

Console.WriteLine("Введите логин: ");

string Login = Convert.ToString(Console.ReadLine());

Console.WriteLine("Введите пароль: ");

string Password = Convert.ToBase64String(ComputeHashSHA256(Encoding.Unicode.GetBytes(Convert.ToString(Console.ReadLine()))));

Console.WriteLine("Регистрация завершена!");

Console.WriteLine("Для входа введите свои учетные данные:");

Console.WriteLine("Введите логин: ");

string EnteredLogin = Convert.ToString(Console.ReadLine());

Console.WriteLine("Введите пароль: ");

string EnteredPassword = Convert.ToBase64String(ComputeHashSHA256(Encoding.Unicode.GetBytes(Convert.ToString(Console.ReadLine()))));

if (Login != EnteredLogin)

{

Console.WriteLine("Введен неверный логин!");

}

else if (Password != EnteredPassword)

{

Console.WriteLine("Введен неверный пароль!");

}

else

{

Console.WriteLine("Авторизация завершена!");

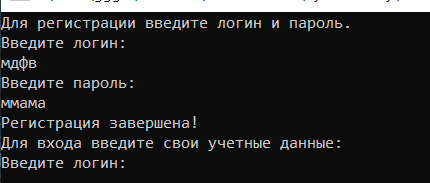
}

}

}

}

(РЕЗУЛЬТАТ)



(ПОСИЛАННЯ)

<https://github.com/VladPonomarenkoi/Cybersecurity>

(ВИСНОВОК)

у ході лабораторної роботи я навчився, використовувати різні алгоритми хешування такі як: MD5, SHA1, SHA256, SHA384, SHA512.