Основи інформаційної безпеки

ПЗ №3

Тема: “Хеш функції та перевірка цілісності інформації”

Підготував:

Студент групи МІТ-21

Міхальський Владислав

Завдання №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("Алгоритм 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("Алгоритм 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("Алгоритм 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("Алгоритм 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("Алгоритм SHA512:");

Console.WriteLine(strForHash1);

Console.WriteLine(strForHash2);

Console.WriteLine(strForHash3);

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

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

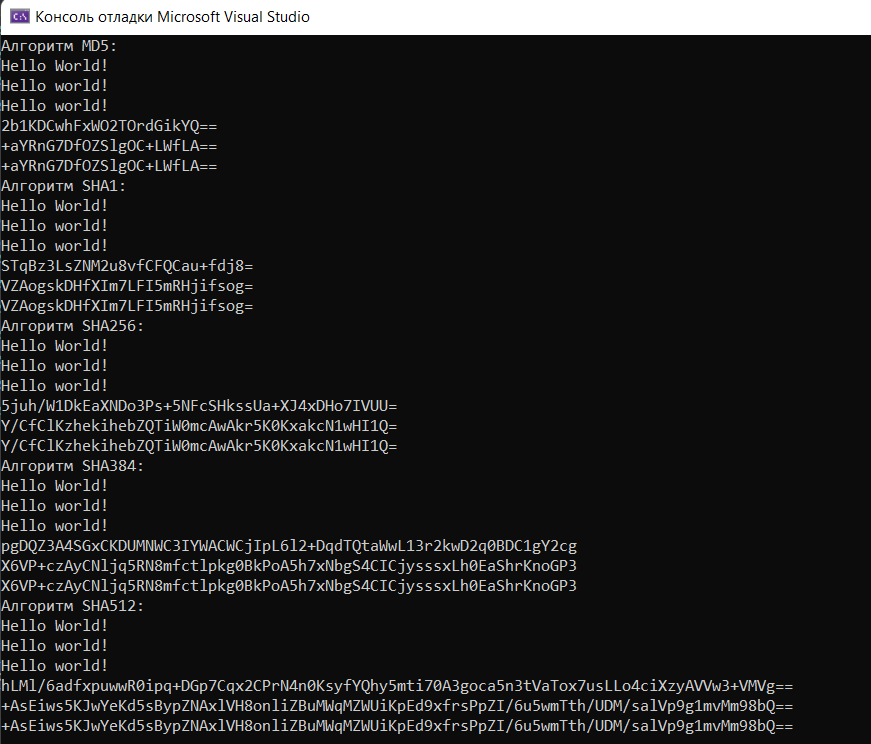
Console.WriteLine(Convert.ToBase64String(sha512ForStr3));

}

}

}

Результат:



Завдання №2

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3\_2

{

class Program

{

static byte[] ComputeHashMd5(byte[] dataForHash)

{

using (var md5 = MD5.Create())

{

return md5.ComputeHash(dataForHash);

}

}

static void Main(string[] args)

{

string hash = "po1MVkAE7IjUUwu61XxgNg==";

for(int i = 100000000; i <= 199999999; i++)

{

var receivedHash = Convert.ToBase64String(ComputeHashMd5(Encoding.Unicode.GetBytes(i.ToString().Substring(1, 8))));

if (receivedHash == hash)

{

Console.WriteLine($"User's password is {i.ToString().Substring(1, 8)}");

}

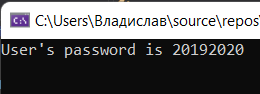
}

}

}

}

Результат:



Завдання №3

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3\_3

{

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 = "Vladyslav123";

string message = "Today is sunny day";

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

var hashedMess = ComputeHmacsha256(Encoding.Unicode.GetBytes(message), hashedKey);

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

Console.WriteLine($"Hashed message: {Convert.ToBase64String(hashedMess)}");

Console.WriteLine("Sending message to receiver....");

string key2 = "Vladyslav123";

var hashedKey2 = ComputeHashSHA256(Encoding.Unicode.GetBytes(key2));

var hashedMess2 = ComputeHmacsha256(Encoding.Unicode.GetBytes(message), hashedKey2);

if (Convert.ToBase64String(hashedMess) == Convert.ToBase64String(hashedMess2))

{

Console.WriteLine("Message is correct!");

}

else

{

Console.WriteLine("Message is broken!");

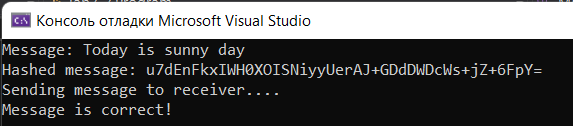
}

}

}

}

Результат:



Завдання №4

using System;

using System.Security.Cryptography;

using System.Text;

namespace lab3\_4

{

class Program

{

static byte[] ComputeHashSHA256(byte[] dataForHash)

{

using (var sha256 = SHA256.Create())

{

return sha256.ComputeHash(dataForHash);

}

}

static void Main(string[] args)

{

Console.WriteLine("For registration enter login and password.");

Console.WriteLine("Enter login: ");

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

Console.WriteLine("Enter password: ");

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

Console.WriteLine("Registration complete!");

Console.WriteLine("To log in, please, enter your credentials:");

Console.WriteLine("Enter login: ");

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

Console.WriteLine("Enter password: ");

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

if (login != enteredLogin)

{

Console.WriteLine("Entered login is incorrect!");

}

else if (password != enteredPassword)

{

Console.WriteLine("Entered password is incorrect!");

}

else

{

Console.WriteLine("Authorization complete!");

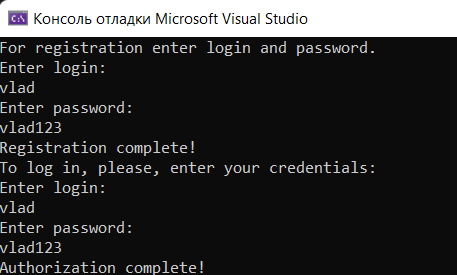
}

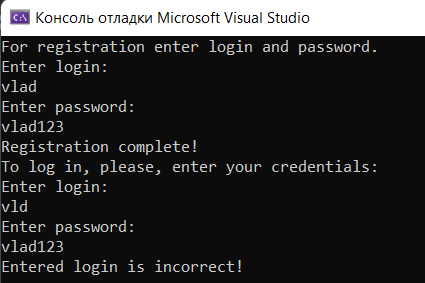
}

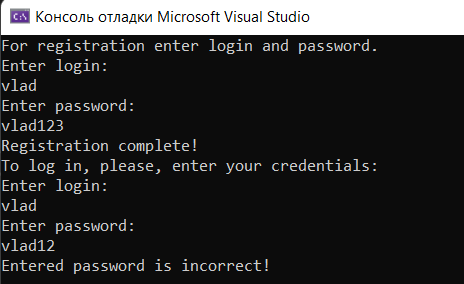
}

}

Результат:







Посилання на Гітхаб репозиторій: https://github.com/vmikhalskyi/Security-basics-2021.git

Висновок: у ході лабораторної роботи я навчився, використовувати різні алгоритми хешування такі як: MD5, SHA1, SHA256, SHA384, SHA512. Навчився перевіряти цілісність інформації завдяки автентифікованому хешуванню.