МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Практическая работа №5**

по дисциплине «Основы информационной безопасности»

на тему: Криптографическая защита информации

Выполнил студент 2 курса 7 группы специальность ПОИБМС Володькин Никифор Дмитриевич

(Ф.И.О.)

Преподаватель Ржеутская Надежда Викентьевна

(Ф.И.О.)

**Цель: изучить и закрепить умение реализации ЭЦП на примере RSA.**

**Выполнение работы.**

**Реализация элементов ЭЦП RSA**

Протоколы ЭЦП с одной стороны относят к протоколам аутентификации, т.к. гарантируют, что сообщение поступило от достоверного отправителя, а с другой стороны к протоколам контроля целостности, т.к. гарантируют, что сообщение пришло в неискаженном виде. Более того, получатель в дальнейшем может использовать ЭЦП как доказательство достоверности сообщения третьим лицам (арбитру) в том случае, если отправитель впоследствии попытается отказаться от него.

Говоря о схеме цифровой подписи, обычно имеют в виду следующую классическую ситуацию:

• отправитель знает содержание сообщения, которое он подписывает;

• получатель, зная открытый ключ проверки подписи, может проверить правильность подписи полученного сообщения в любое время без какого-либо разрешения и участия отправителя;

• безопасность схемы подписи гарантируется.

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (Федеральный закон "Об электронной цифровой подписи").

При создании цифровой подписи по классической схеме отправитель:

• применяет к исходному сообщению T хеш-функцию h(T) и получает хеш-образ r сообщения;

• вычисляет цифровую подпись s по хеш-образу r с использованием своего закрытого ключа;

• посылает сообщение T вместе с цифровой подписью s получателю.

Получатель, отделив цифровую подпись от сообщения, выполняет следующие действия:

• применяет к полученному сообщению T хеш-функцию h(T) и получает хеш-образ r сообщения;

• расшифровывает хеш-образ r’ из цифровой подписи s с использованием открытого ключа отправителя;

• проверяет соответствие хеш-образов r и r’ и если они совпадают, то отправитель действительно является тем, за кого себя выдает, и сообщение при передаче не подверглось искажению.

Как видно из этой схемы, порядок использования ключей обратный тому, который используется при передаче секретных сообщений. Вначале отправитель использует свой закрытый ключ, а затем получатель применяет открытый ключ отправителя.

Разновидности ЭЦП

Кроме классической схемы ЭЦП различают еще несколько специальных:

• схема "конфиденциальной" (неотвергаемой) подписи – подпись не может быть проверена без участия сгенерировавшего ее лица;

• схема подписи "вслепую" ("затемненной" подписи) - отправитель не знает подписанного им сообщения;

• схема "мультиподписи" - вместо одного отправителя сообщение подписывает группа из нескольких участников;

• схема "групповой" подписи - получатель может проверить, что подписанное сообщение пришло от члена некоторой группы отправителей, но не знает, кем именно из членов группы оно подписано. В тоже время, в случае необходимости, отправитель может быть определен;

• и др.

Этап 1. Выработка ключей (выполняет отправитель А) - см. практическую работу 6 "Шифрование методом RSA".

Этап 2. Отправка сообщения и электронной подписи (выполняет отправитель А).

Отправка сообщения и ЭЦП на базе алгоритма RSA



Этап 3. Получение сообщения и проверка электронной подписи (выполняет получатель B).



Получение сообщения и проверка ЭЦП на базе алгоритма RSA.

Было написано следующее приложение на языке C#, реализующее ЭЦП RSA с помощью класса **RSACryptoServiceProvider**:

using System;

using System.Security.Cryptography;

using System.Text;

class RSASign

{

static void Main()

{

try

{

UTF8Encoding ByteConverter = new UTF8Encoding();

Console.Write("Введите сообщение: ");

string dataString = Console.ReadLine();

byte[] originalData = ByteConverter.GetBytes(dataString);

byte[] signedData;

RSACryptoServiceProvider RSAalg = new RSACryptoServiceProvider();

RSAParameters Key = RSAalg.ExportParameters(true);

signedData = HashAndSignBytes(originalData, Key);

if (VerifySignedHash(originalData, signedData, Key))

{

Console.WriteLine("ЭЦП принадлежит этому сообщению");

}

else

{

Console.WriteLine("ЭЦП не принадлежит этому сообщению");

}

}

catch (ArgumentNullException)

{

Console.WriteLine("ЭЦП не проверено");

}

}

public static byte[] HashAndSignBytes(byte[] DataToSign, RSAParameters Key)

{

try

{

RSACryptoServiceProvider RSAalg = new RSACryptoServiceProvider();

RSAalg.ImportParameters(Key);

return RSAalg.SignData(DataToSign, SHA256.Create());

}

catch (CryptographicException e)

{

Console.WriteLine(e.Message);

return null;

}

}

public static bool VerifySignedHash(byte[] DataToVerify, byte[] SignedData, RSAParameters Key)

{

try

{

RSACryptoServiceProvider RSAalg = new RSACryptoServiceProvider();

RSAalg.ImportParameters(Key);

return RSAalg.VerifyData(DataToVerify, SHA256.Create(), SignedData);

}

catch (CryptographicException e)

{

Console.WriteLine(e.Message);

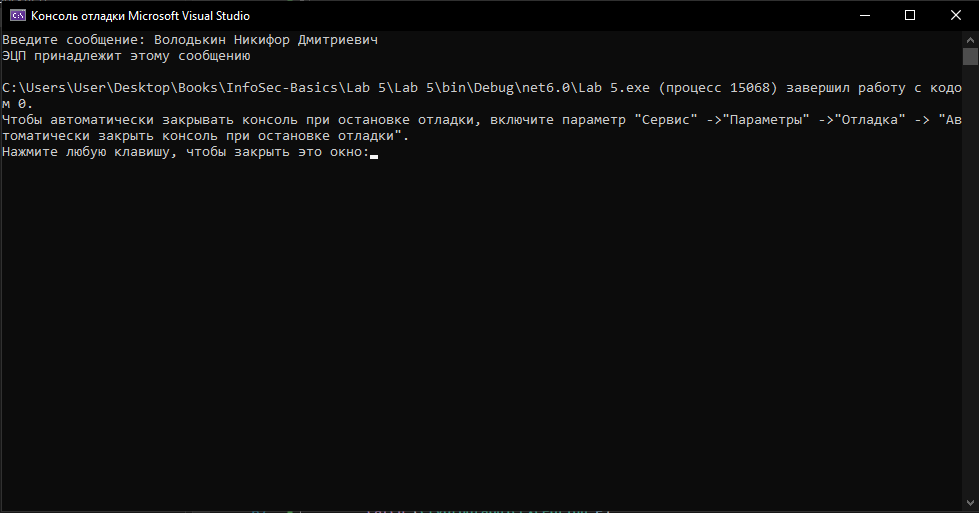
return false;

}

}

}

Результат выполнения программы:



**Вывод: изучены и закреплены умения реализации ЭЦП.**